Final Report

Grant No DE-FG02-05ER25667

Adaptive LES Methodology for Turbulent Flow Simulations

by

Oleg V. Vasilyev

June 2008

Contents

1 Introduction

Although turbulent flows are common in the world around us, a solution to the fundamental equations that govern turbulence still eludes the scientific community. Turbulence has often been called one of the last unsolved problem in classical physics, yet it is clear that the need to accurately predict the effect of turbulent flows impacts virtually every field of science and engineering. As an example, a critical step in making modern computational tools useful in designing aircraft is to be able to accurately predict the lift, drag, and other aerodynamic characteristics in numerical simulations in a reasonable amount of time. Simulations that take months to years to complete are much less useful to the design cycle. Much work has been done toward this goal (Lee-Rausch *et al.* 2003, Jameson 2003) and as cost effective accurate tools for simulating turbulent flows evolve, we will all benefit from new scientific and engineering breakthroughs.

The problem of simulating high Reynolds number (Re) turbulent flows of engineering and scientific interest would have been solved with the advent of Direct Numerical Simulation (DNS) techniques if unlimited computing power, memory, and time could be applied to each particular problem. Yet, given the current and near future computational resources that exist and a reasonable limit on the amount of time an engineer or scientist can wait for a result, the DNS technique will not be useful for more than "unit" problems for the foreseeable future (Moin & Kim 1997, Jimenez & Moin 1991). The high computational cost for the DNS of three dimensional turbulent flows results from the fact that they have eddies of significant energy in a range of scales from the characteristic length scale of the flow all the way down to the Kolmogorov length scale. The actual cost of doing a three dimensional DNS scales as $\mathrm{Re}^{9/4}$ due to the large disparity in scales that need to be fully resolved. State-of-the-art DNS calculations of isotropic turbulence have recently been completed at the Japanese Earth Simulator (Yokokawa *et al.* 2002, Kaneda *et al.* 2003) using a resolution of 4096^3 (approximately 10^{11}) grid points with a Taylor-scale Reynolds number of 1217 ($\mathrm{Re} \approx 10^6$). Impressive as these calculations are, performed on one of the world's fastest super computers, more brute computational power would be needed to simulate the flow over the fuselage of a commercial aircraft at cruising speed. Such a calculation would require on the order of 10^{16} grid points and would have a Reynolds number in the range of 10^8. Such a calculation would take several thousand years to simulate one minute of flight time on today's fastest super computers (Moin & Kim 1997). Even using state-of-the-art zonal approaches, which allow DNS calculations that resolve the necessary range of scales within predefined "zones" in the flow domain, this calculation would take far too long for the result to be of engineering interest when it is finally obtained.

Since computing power, memory, and time are all scarce resources, the problem of simulating turbulent flows has become one of how to abstract or simplify the complexity of the physics represented in the full Navier-Stokes (NS) equations in such a way that the "important" physics of the problem is captured at a lower cost. To do this, a portion of the modes of the turbulent flow field needs to be approximated by a low order model that is cheaper than the full NS calculation. This model can then be used along with a numerical simulation of the "important" modes of the problem that cannot be well represented by the model. The decision of what part of the physics to model and what kind of model to use has to be based on what physical properties are considered "important" for the problem. It should be noted that "nothing is free", so any use of a low order model will by definition lose some information about the original flow.

In an abstract sense the first question to address, when one is looking to develop a reduced order method for simulating turbulent flows, is how to determine what part of the physical system will be approximated with a low order model and what part will be simulated numerically. One choice is to average the behavior of the system over time. This technique is used in Reynolds Averaged Navier Stokes (RANS) simulations (Durbin & Reif 2001, Pope 2000, Gatski *et al.* 1996). The main limitation of RANS approach is that it is highly empirical and sensitive to the model parameters that have to be tuned to the problem.

Another option is to simulate the flow on an adapted grid that is coarser than the grid necessary to represent the flow down to the Kolmogorov length scale. The goal of this type of method is to resolve the energetic eddies that dominate the flow physics. Any coarsening of the grid, either locally or globally, implies that not all the modes or frequencies of the original flow are resolved. Therefore, these "missing" modes will have to be modeled somehow. We will refer to this class of methods as Eddy Capturing Methods.

The dominant method in the class of Eddy Capturing Methods is Large Eddy Simulation (LES), where the formal scale separation is obtained by means of a low-pass filtering operation applied to the Navier-Stokes equations, which leads to the definition of filtered (or large-scale) and residual (or small-scale) fields. The filtered Navier-Stokes equations are closed by modeling the subgrid-scale (SGS) stresses that account for the effect of the unresolved small-scale eddies. In order to realize the benefit of LES, a low order model for the SGS stress, which is based on the resolved quantities, is needed. In practice τ_{ij} can be modeled either deterministically (Smagorinsky 1963, Bardina *et al.* 1983, Germano *et al.* 1991, Lesieur & Métais 1996, Meneveau & Katz 2000, Moin 2002) or stochastically (Chasnov 1991). The inherent problem with this working definition for LES is that it resolves the large scale eddies instead of the coherent energetic eddies. It has been shown that the coherent energetic eddies in a turbulent flow contain significant energy at all length scales from the characteristic length scale of the domain down to the Kolmogorov length scale (Jimenez *et al.* 1993, Goldstein *et al.* 2000, Farge *et al.* 2001, Farge & Schneider 2001). Vorticity tubes, which seem to be the basic structure of three-dimensional homogeneous turbulence, are observed at all scales of the flow (Vincent & Meneguzzi 1991). Therefore, when a spectral cutoff filter or low-pass filter with grid truncation is used with LES, the small scale structure of the coherent energetic eddies are not resolved. Another problem with LES is that the computational grid is commonly defined *a priori*, based on the physics and geometry of the problem (Moin 2002, Piomelli 1999, Wang & Moin 2002). Yet in flows of engineering and scientific importance the large scales of interest often change over the domain of the problem and in time. As stated in the recent book on turbulence by Pope (Pope 2000), "the ideal numerical method for LES would include adaptive gridding to ensure automatically that the grid, and hence the filter, are everywhere sufficiently fine to resolve the energy-containing motions". This implies that there is a need for an Eddy Capturing Method that is adaptive in time and space.

Turbulence is characterized by energetic eddies that are localized in space and scale, yet the methods discussed so far do not take advantage of this localization. There exists the possibility of exploiting this localization by "compressing" the turbulence problem such that a simulation with a subset of the total modes captures the dynamics of the most energetic eddies in the flow. A new method for simulating turbulence called Coherent Vortex Simulation, was recently introduced by Farge *et al.* (Farge *et al.* 1999). This method uses a wavelet filter to dynamically resolve and "track" the energetic coherent eddies or vortices in a turbulent flow. It has been shown that the resulting SGS field with CVS is near Gaussian white noise (Goldstein *et al.* 2003, Schneider *et al.* 2003). Therefore, if only low order statistics such as SGS dissipation are required, a CVS simulation can be run with no subgrid scale model. This is possible because a purely incoherent white noise SGS field will result in zero total SGS dissipation from the resolved field. It is important to note that there is still significant energy transfer between the resolved and SGS modes and viceversa, but the statistical average or net energy transfer is zero. If higher order statistics are required, then a purely stochastic subgrid stress model should be used to reproduce the effect of the subgrid scales. The use of the stochastic model has not been explored up-to-now in CVS. One of the challenges with the CVS method is how to determine on the fly during an actual simulation the "ideal" wavelet compression, which results in a purely incoherent subgrid scale field. Even if it can be found in a cost effective manner, it is still likely that the associated adaptive grid will be too fine to be cost effective for simulating high *Re* number flows, since the computational cost of CVS falls between DNS and LES.

The main objective of this research project is to develop the adaptive LES methodology. The new approach inherits the advantages of both the CVS and LES methods: the ability of the CVS method to dynamically resolve and "track" the most energetic part of the coherent eddies in a turbulent flow field with a field compression in the range of that used with typical LES applications. In this project, both the CVS and adaptive LES methods are implemented using a Dynamically Adaptive Wavelet Collocation method (DAWCM) (Vasilyev & Bowman 2000, Vasilyev 2003). The DAWCM is ideal for CVS and adaptive LES as it combines the resolution of the energetic coherent modes in a turbulent flow with the simulation of their temporal evolution (Vasilyev & Kevlahan 2002, Kevlahan *et al.* 2003, Goldstein *et al.* 2003, Kevlahan & Vasilyev 2005). The wavelet collocation method employs wavelet compression as an integral part of the solution such that the solution is obtained with the minimum number of grid points for a given accuracy. When the threshold is chosen simply to satisfy numerical accuracy (and subgrid scales are not modeled) we call this method Wavelet based Direct Numerical Simulation, or WDNS.

During the duration of the project our efforts were concentrated in two main areas: model and algorithm development. The main accomplishments in both of these areas are summarized below:

Model Development

1. The new **global** dynamic SGS stress modeling procedure based on a variation of the classical Smagorinsky (Smagorinsky 1963) model is developed (Goldstein *et al.* 2005). In this model the scaling of the eddy viscosity is based on ϵ^2 (where ϵ is the wavelet filtering threshold parameter), instead of the standard scaling $\overline{\Delta}^2$ (where Δ is the filter width). The model uses global dynamic coefficient. The model has been successfully tested for decaying homogeneous turbulence case. The detailed discussion is presented in Sections 4, 7.1-7.3.
2. New local SCALES models based on Lagrangian path-line diffusive and path-tube averaging are developed (De Stefano *et al.* 2005, Vasilyev *et al.* 2006, Vasilyev *et al.* 2008) and assessed in terms of accuracy and efficiency. The models are tested for freely decaying homogeneous turbulence with initial $Re_\lambda = 72$. It is shown that the SCALES results, obtained with fewer than 0.4% of the total non-adaptive nodes required for a DNS with the same wavelet solver, closely match reference DNS data. In contrast to classical LES, this agreement holds not only for large scale global statistical quantities, but also for energy and, more importantly, enstrophy spectra up to the dissipative wavenumber range. The detailed discussion is presented in Sections 5.1 and 7.4.
3. New localized dynamic models for stochastic coherent adaptive large eddy simulation that involve an evolution equation for the subgrid kinetic energy are developed (Vasilyev *et al.* 2006, De Stefano *et al.* 2008). One of the main advantages of this formulation is that the equilibrium assumption between production and dissipation of SGS energy is not required as in the classical Smagorinsky approach. In contrast, the energy transfer between resolved and residual motions is directly ensured by solving an additional transport model equation for the subgrid-scale energy. Some known difficulties, associated with the classical dynamic Germano model are overcome using these models. Specifically, scaling subgrid scale stress in terms of the SGS kinetic energy provides a feedback mechanism that makes the numerical simulation stable regardless of whether an eddy-viscosity or non-eddy viscosity assumption is made. This way, no averaging procedure is needed in practice and the models stay fully localized in space. The detailed discussion is presented in Sections 5.2, 5.3, 7.4, 7.5, and 7.6.

Algorithm Development

1. Three different data-structures have been developed: working array, Lines, and Trees. All three data-structures are implemented and all solver features are extended to these data-structures. The preliminary result demonstrate that lines-based data-structure is more efficient in terms of cash locality. Despite computational savings, the Lines data-structure adds substantial technical complexity into the algorithm by only having access to the nearest points along the line, which makes this data structure impractical for development purposes. The detailed discussion of data-structures is presented in Section 6.1.
2. A novel modified asynchronous adaptive wavelet transform that completely eliminated the requirement of level synchronization is developed. The new algorithm completely eliminated the requirement of synchronizing data between different processors for each level of wavelet transform, which it turn puts additional requirement on load balancing of each level of resolution. Instead both the wavelet transform and/or linear operator can be evaluated in each computational subdomain with only one inter-processor synchronization for each operation. The only disadvantage of such algorithm is the loss of the zero-mean property of the wavelets in the vicinity of the inter-processor domain boundary. This drawback is easily compensated by the simplicity of the load balancing algorithm and possible increase of local resolution in the immediate proximity of the sub-domain boundary. The detailed discussion of the modified asynchronous wavelet transform is presented in Section 6.2.
3. A parallel version of the code has been developed. The code was designed to consist of three main parts: the main wavelet code, the data structure, and the parallel communicators with the required interfaces and external libraries. In addition, all the information transfers between processors were implemented via collective all-to-all MPI (message passing interface) communications, which eliminated all the deadlock related problems. The major trade-off of such an approach was, again, the code performance. Normally, a processor would require information from several neighboring domains only, not from all the other processors, therefore replacing all-to-all with point-to-point communications is an important optimization which will be implemented in the future. The parallel solver has been assessed in terms of scalability and efficiency and the ares for future improvements of the solver were identified. The detailed discussion of parallelization issues and load balancing algorithm Section 6.3.

The rest of the report is organized as follows. A brief review of the background relevant to this progress report is given in Section 2. The Stochastic Coherent Adaptive Large Eddy Simulation (SCALES) approach is formulated in Section 3. The global dynamic model for SCALES is presented in Section 4. The local dynamic models for SCALES are presented in Section 5. The issues related to algorithm development, data structures, and load balancing procedure are discussed in Section 6. The results of CVS and SCALES simulaitons with global and local dynamic models for decaying isotropic turbulent flow are presented in Section 7. Finally the main achievements and possible future extension of the work are discussed in detail in Section 8.

2 Background

In the following subsections we briefly discuss background theory relevant to this progress report. First the Large Eddy Simulation method is introduced followed by the discussion of the two important properties of wavelets (Sweldens 1996, Sweldens 1998) that are essential for this project, namely the ability of wavelets to compress and de-noise signals. Then we will briefly describe dynamically adaptive wavelet collocation method that is used as a base method for this project.

2.1 Large Eddy Simulation

The large eddy simulation (LES) method is based on the premise that the large scales of a turbulent flow dominate mixing, heat transfer and other quantities of engineering interest, while the small scales are only of interest because of how they effect the large scales. The LES equations for incompressible flow, that describe the evolution of the large scale eddies in the flow field, can be written as:

$$\frac{\partial \overline{u_i}}{\partial x_i} = 0\,, \tag{1}$$

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial (\overline{u}_i\,\overline{u}_j)}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \nu \frac{\partial^2 \overline{u}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j}\,, \tag{2}$$

where

$$\tau_{ij} = \overline{u_i u_j} - \overline{u}_i\,\overline{u}_j \tag{3}$$

and u_i is the velocity field, ρ is density, ν is kinematic viscosity, p is pressure and $\overline{(\cdot)}$ represents spatial filtering. As a result of the filtering process the unresolved quantity τ_{ij}, commonly referred to as the subgrid scale (SGS) stress, is introduced. Note that τ_{ij} is a function of the unfiltered velocity field u_i. In order to close (1–2) and realize the benefit of LES, a low order model for the SGS stress, which is based on the resolved quantities, is needed. In practice τ_{ij} can be modeled either deterministically (Moin 2002, Lesieur & Métais 1996, Meneveau & Katz 2000) or stochastically (Chasnov 1991). Most current LES is done using purely deterministic models of the eddy viscosity type (Pope 2000).

In LES the filter is either explicit or it can be defined implicitly by the computational grid. Either way, LES uses a reduced computational grid that is capable of supporting (or representing) only a subset of the total number of active modes in the flow. Current state of the art LES work uses non-uniformly stretched meshes or zonal grids (Moin 2002, Piomelli 1999, Kravchenko *et al.* 1996) that are refined *a priori* to the geometry of the problem.

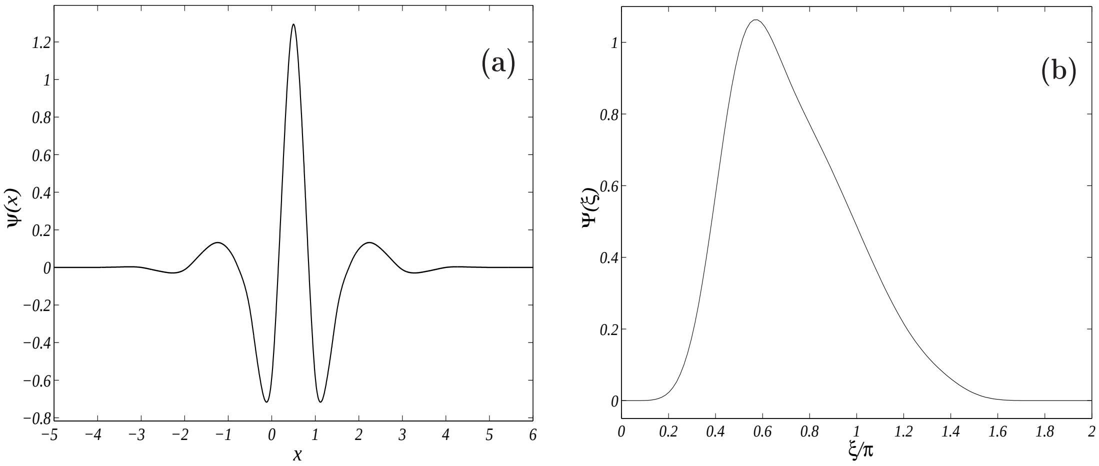


Figure 1. Lifted interpolating wavelet ψ, of order 6 (a) and its Fourier transform $\Psi(\xi)$ (b).

2.2 General Properties of Wavelets

Wavelets are basis functions which are localized in both physical space (due to their finite support) and wavenumber space (due to their vanishing moments), *e.g.* Fig. 1. For comparison, the classical Fourier transform is based on functions (sines and cosines) that are well localized in wavenumber, but do not provide localization in physical space due to their global support. Because of this space/scale localization, the wavelet transform provides both spatial and scale (frequency) information while the Fourier transform on the other hand only provides frequency information.

A scalar field $f(\mathbf{x})$ can be represented in terms of wavelet basis functions as

$$f(\mathbf{x}) = \sum_{\mathbf{l}\in\mathcal{L}^0} c^0_{\mathbf{l}}\phi^0_{\mathbf{l}}(\mathbf{x}) + \sum_{j=0}^{+\infty}\sum_{\mu=1}^{2^n-1}\sum_{\mathbf{k}\in\mathcal{K}^{\mu,j}} d^{\mu,j}_{\mathbf{k}}\psi^{\mu,j}_{\mathbf{k}}(\mathbf{x}) , \tag{4}$$

where $\phi^0_{\mathbf{k}}(\mathbf{x})$ and $\psi^{\mu,j}_{\mathbf{l}}$ are respectively n-dimensional scaling functions and wavelets of different families (μ) and levels of resolution (j). One may think of a wavelet decomposition as a multilevel or multiresolution representation of a function, where each level of resolution j (except the coarsest one) consists of wavelets $\psi^j_{\mathbf{l}}$ or family of wavelets $\psi^{\mu,j}_{\mathbf{l}}$ having the same scale but located at different positions. Scaling function coefficients represent the averaged values of the field, while the wavelet coefficients represent the details of the field at different scales. The wavelet functions have a zero mean, while the scaling functions do not. Note that in n-dimensions there are $2^n - 1$ distinctive n-dimensional wavelets (Daubechies 1992). Also note that due to the local support of both scaling functions and wavelets, there is a one-to-one correspondence between the location of each scaling function or wavelet with a grid point. As a result each scaling function coefficient $c^0_{\mathbf{l}}$ and each wavelet coefficient $d^{\mu,j}_{\mathbf{k}}$ is uniquely associated with a single grid point with the indices $\mathbf{l}$ and $\mathbf{k}$ respectively.

Traditionally, one dimensional first generation wavelets ψ^j_k are defined as translates and dilates of one basic wavelet ψ, *i.e.* $\psi^j_k(x) = \psi(2^j x - k)$. Second generation wavelets (Sweldens 1996, Sweldens 1998) are a generalization of first generation wavelets that supplies additional freedom to deal with arbitrary boundary conditions, and irregular sampling intervals. Second generation wavelets form a Riesz basis for $\mathbf{L}_2$ space, with the wavelets being local in both space and frequency and often having many vanishing polynomial moments, but without the translation and dilation invariance of their first generation cousins. Despite the loss of these two fundamental properties of wavelet bases, second generation wavelets retain many of the useful features of first generation wavelets, including a fast $O(N)$ transform. The construction of second generation wavelets is based on the lifting scheme that is discussed in detail by Sweldens (Sweldens 1996, Sweldens 1998).

For this project we use a set of second generation wavelets known in the literature as lifted interpolating wavelets (Vasilyev & Bowman 2000, Sweldens 1996). In particular, simulations with the dynamically adaptive wavelet collocation (DAWC) solver are run using a lifted interpolating wavelet of order 6, which is shown in Fig. 1 along with its Fourier transform. For a more in-depth discussion on the construction of these wavelets the reader is referred to the papers by Sweldens (Sweldens 1996, Sweldens 1998), and Vasilyev and Bowman (Vasilyev & Bowman 2000). For a more general discussion on wavelets we refer the reader to the books of Daubechies (Daubechies 1992) and Mallat (Mallat 1999).

2.3 Wavelet Filters

Wavelet filtering is performed in wavelet space using wavelet coefficient thresholding, which can be considered as a nonlinear filter that depends on each flow realization. The wavelet thresholding filter is defined by,

$$\overline{f}^{>\epsilon}(\mathbf{x}) = \sum_{\mathbf{l}\in\mathcal{L}^0} c^0_{\mathbf{l}}\phi^0_{\mathbf{l}}(\mathbf{x}) + \sum_{j=0}^{+\infty}\sum_{\mu=1}^{2^n-1} \sum_{\substack{\mathbf{k}\,\in\,\mathcal{K}^{\mu,j}\\ |d^{\mu,j}_{\mathbf{k}}|\,>\,\epsilon\|f\|_{\text{WTF}}}} d^{\mu,j}_{\mathbf{k}}\psi^{\mu,j}_{\mathbf{k}}(\mathbf{x}) \,, \tag{5}$$

where $f(\mathbf{x})$ is a scalar field, $\epsilon > 0$ stands for the non-dimensional (relative) threshold parameter, and $\|\cdot\|_{\text{WTF}}$ being the Wavelet Threshold Filtering (WTF) norm that provides the (absolute) dimensional scaling for filtered variable f. For instance, in the case of velocity, the (absolute) dimensional scaling can be specified as the L_2 norm ($\|u_i\|_{\text{WTF}} = \|u_i\|_2$) or the L_∞ norm ($\|u_i\|_{\text{WTF}} = \|u_i\|_\infty$). Note that once the WTF-norm $\|\cdot\|_{\text{WTF}}$ is specified, the wavelet thresholding filter (5) is **uniquely** defined by the nondimensional threshold parameter, ϵ.

The reconstruction error due to wavelet filtering with non-dimensional threshold parameter ϵ can be shown to be (Donoho 1992, Vasilyev 2003):

$$\|f(\mathbf{x}) - \overline{f}^{>\epsilon}(\mathbf{x})\|_2 \leq C\epsilon\|f\|_{\text{WTF}} \,, \tag{6}$$

for a piecewise regular function $f(\mathbf{x})$, where C is of order unity.

As will be shown later, when the wavelet threshold filter is applied to a system of evolution equations, each variable could be filtered according to Eq. (5). Once filtered, each variable could be integrated in time. However, this would lead to numerical complications due the one-to-one correspondence between the location of a wavelet with a grid point. In particular each variable would be solved on a different grid. In order to avoid this difficulty and make filtering of each term in the evolution equation easy, in the present study the coupled wavelet thresholding strategy is used. The mask of significant wavelet coefficients is constructed for each variable according to thresholding criteria of Eq. (5). The union of these masks will result in the *global* thresholding mask, that is used for each dependent variable and each term in the equation. Note that in some applications additional variables, like vorticity or strain rate can be used for construction of the *global* mask. Once this *global* mask is constructed, one can view the wavelet filtering as local low-pass filtering, where the high frequencies are removed according to the *global* mask. The effective wavelet filter width depends on the choice of WTF-norm, the spatial distribution of the variables used for defining the coupled wavelet filter mask and is a function of non-dimensional threshold parameter, ϵ. Such interpretation of wavelet threshold filtering highlights the similarity between SCALES and classical LES approaches. However, the wavelet thresholding filter is drastically different from the LES filters, primarily because it changes in time following the evolution of the solution, which, in turn, results in an adaptive computational grid that tracks in physical space the areas of locally significant energy of all variables used for the grid adaptation. However, it is important to note that, unlike the Fourier modes, there is no one-to-one correspondence between wave number and wavelet level. Instead, each wavelet level represents a region of wave numbers. Figure 2 shows the energy spectra of the modes associated with 6 wavelet scales or levels along with the full energy spectra of a turbulent field, obtained from a 256^3 DNS simulation of forced isotropic turbulence (Jimenez *et al.* 1993) with $\mathrm{Re}_\lambda = 168$. Note that this turbulent field will hereafter be referred to as F_{256}. This figure highlights the fact that each wavelet scale has energy in a region of wave numbers, and that these regions overlap.

2.4 Wavelet Compression and Wavelet De-noising

The major strength of wavelet filtering decomposition (5), is the ability to compress signals. For functions that contain isolated small scales on a large-scale background, most wavelet coefficients are small, thus, we can retain good approximation even after discarding a large number of wavelets with small coefficients. Intuitively, the coefficient $d_{\mathbf{l}}^{\mu,j}$ will be small unless $u(\mathbf{x})$ has variation on the scale of j in the immediate vicinity of wavelet $\psi_{\mathbf{l}}^{\mu,j}(\mathbf{x})$.

Another important property of wavelet analysis used in this project is the ability of wavelets to de-noise signals. The wavelet de-noising procedure, also called wavelet-shrinkage,

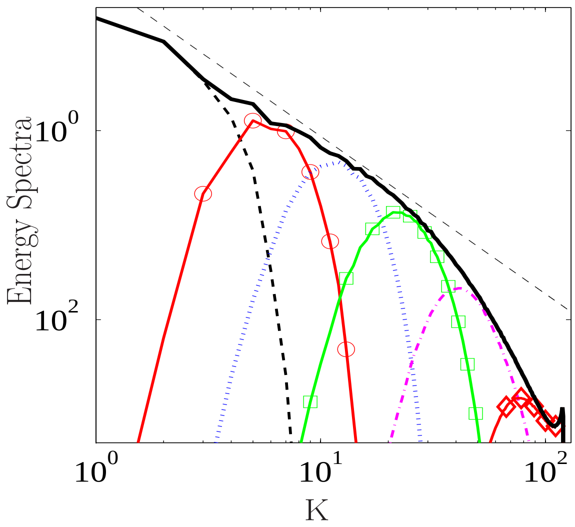


Figure 2: Energy spectra of turbulent velocity field F_{256}: (▬▬), contained in wavelet levels. Level 1: (- - - -), Level 2: (—○—), Level 3: (——), Level 4: (——), Level 5: (—·—), Level 6: (—◇—)

was introduced by Donoho (Donoho 1993, Donoho 1994) based on orthogonal wavelet decompositions. It can be described as follows: given a function that consists of a smooth function with superimposed noise, one performs a forward wavelet transform and sets to zero "noisy" wavelet coefficients (i.e. those wavelet coefficients whose modulus squared is less than the noise variance σ^2), otherwise the wavelet coefficient is kept. This procedure is known as hard thresholding. Donoho (Donoho 1993) demonstrated that hard thresholding is optimal for de-noising signals in the presence of Gaussian white noise. In the CVS method discussed in this report the "noise" is actually the SGS modes.

2.5 Dynamically Adaptive Wavelet Collocation Method (DAWCM)

A key component in the implementation of the SCALES method is the development of a dynamically adaptive wavelet collocation (DAWC) solver (Vasilyev 2003, Vasilyev & Bowman 2000, Kevlahan & Vasilyev 2005, Vasilyev & Paolucci 1997). This solver is ideally suited to the simulation of turbulence since wavelets adapt the numerical resolution naturally to the localized turbulent structures that exist at all wave numbers in fully developed turbulence. The wavelet collocation method takes advantage of the fact that wavelets are localized in both space and scale, and as a result, functions with localized regions of sharp transition are well compressed using wavelet decomposition. The adaptation is achieved by retaining only those wavelets, whose coefficients satisfy the thresholding criteria of Eq. (5). Thus, high resolution computations are carried out only in those regions, where sharp transitions occur. With this adaptation strategy, a solution is obtained on a near optimal grid that "tracks" the coherent vortices in the field, *i.e.* far fewer grid points are needed for

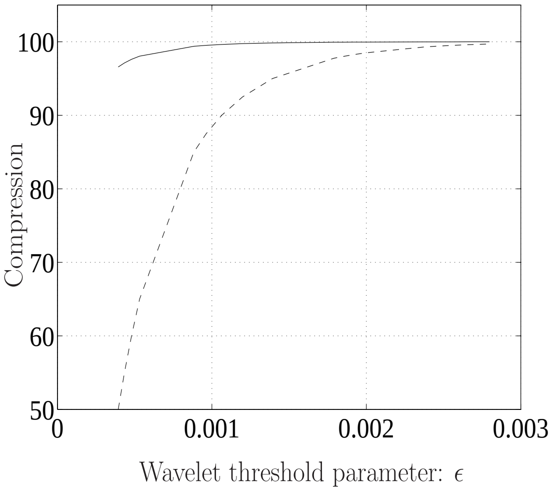


Figure 3: Field compression vs. relative wavelet threshold parameter, ϵ, using velocity wavelet filtering, without adjacent zone (———) and with adjacent zone (- - - -) for field F_{256}. It can be seen that as ϵ increases the loss in compression due to the adjacent zone becomes less significant. The (absolute) dimensional scaling $\|u_i\|_{\mathrm{WTF}} = \|u_i\|_\infty$ was used in the wavelet filtering for this *a priori* test.

wavelets than for conventional finite-difference, finite-element, or spectral methods (Farge 1992). By varying the threshold parameter ϵ this method can be used to implement any of the wavelet based methods discussed above, namely WDNS, CVS or SCALES. The dynamically adaptive wavelet collocation algorithm has already been successfully applied to the solution of thermo-acoustic wave propagation problems (Vasilyev & Paolucci 1997), combustion problems (Vasilyev 2003, Vasilyev & Bowman 2000), fluid–structure interaction problems (Kevlahan & Vasilyev 2005), viscoelastic flows (Vasilyev *et al.* 1997, Vasilyev *et al.* 2001), and the compaction phenomenon in a poro-viscoelastic matrix (Vasilyev *et al.* 1998).

Let us briefly outline the main features of the numerical method. Details can be found in Refs. (Vasilyev 2003, Vasilyev & Bowman 2000). In the wavelet collocation method there is a one-to-one correspondence between grid points and wavelets. This makes calculation of nonlinear terms simple, and allows the grid to adapt automatically to the solution at each time step by adding or removing wavelets. Very briefly, at each time step we take the wavelet transform of the solution and apply the *global* thresholding mask to remove wavelets, which do not satisfy the thresholding criteria of Eq. (5) for all of the adaptation variables. To account for the evolution of the solution over one time step the computational grid needs to be extended to include grid points associated with wavelets whose coefficients are, or can possibly become, significant during the time integration step (Liandrat & Tchamitchian 1990). To do this we add grid points that are adjacent in both position and scale to each significant wavelet coefficient. While the cost of this added adjacent zone is significant at low compression ratios it becomes much less so at higher compression ratios. This diminishing cost of the adjacent zone with increased compression will be the case for any numerical problem that has inherent local structures that dominate the field being simulated. Figure 3 shows the compressifon ratio vs. the wavelet filter threshold parameter ϵ for a wavelet collocation grid adapted to a DNS field of isotropic turbulence ($\mathrm{Re}_\lambda = 168$) with and without an adjacent zone. For this *a priori* test the coupled wavelet filter was applied based on wavelet thresholding of the velocity components using L_∞ WTF-norm. We can see clearly that the added overhead of the adjacent zone becomes insignificant for compression ratios over 98%. This is the case because in turbulent flows, like the one considered, the flow is dominated by localized energetic coherent vortices. This trend will also hold for other common flows such as flow fields involving vortices due to fluid–structure interaction or shocks in compressible flow fields. Since each wavelet corresponds to a single grid point this procedure allows the grid to automatically follow the evolution of the solution in position and scale. We use second generation wavelets (Sweldens 1998), which allow the order of the wavelet (and hence of the numerical method) to be varied easily. The method has a computational complexity $O(N)$, where N is the number of wavelets retained in the calculation (*i.e.* the union of those wavelets with coefficients greater than $\epsilon\|f_i\|_{\mathrm{WTF}}$, for all adaptation variables, plus nearest neighbors).

In summary, the dynamically adaptive wavelet collocation method is an adaptive, variable order method for solving partial differential equations with localized structures that change their location and scale in space and time. Because the computational grid automatically adapts to the solution (in position and scale), we do not have to know *a priori* where the regions of high gradients or structures exist.

2.5.1 DAWCM in Complex Geometry

In order to solve problems in geometries of engineering interest, the DAWCM employs Brinkman penalization (Khadra *et al.* 2000) as a way of simulating the presence of arbitrarily complex solid boundaries (which may be moving in time). The Brinkman penalization technique allows boundary conditions to be enforced to a specified precision, without changing the numerical method (or grid) used to solve the equations. The main advantage of this method, compared to other penalization type methods, is that the error can be estimated rigorously in terms of the penalization parameter. It can also be shown that the solution of the penalized equations converges to the exact solution in the limit as the penalization parameter tends to zero (Khadra *et al.* 2000).

Let us briefly outline the main features of the numerical method on the example of incompressible Navier-Stokes equations. We consider here the case of the flow around a number of obstacles O_i. The problem is solved in a rectangular domain containing all the obstacles O_i. To model the effect of the no-slip boundary conditions on the obstacles O_i without explicitly imposing them on the boundaries of the obstacles we solve the following

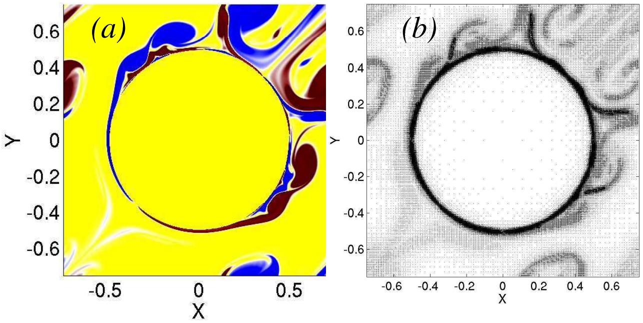


Figure 4: Vorticity field (a) and corresponding computational grid (b) for the direct numerical simulation of flow around two-dimensional periodic cylinder array at $Re = 10^4$ using DAWCM.

set of *penalized* equations

$$\begin{aligned} \frac{\partial \boldsymbol{u}_\eta}{\partial t} + \boldsymbol{u}_\eta \cdot \nabla \boldsymbol{u}_\eta + \nabla P_\eta &= \nu \Delta \boldsymbol{u}_\eta - \frac{1}{\eta}\chi_0 \boldsymbol{u}_\eta, && (7)\\ \nabla \cdot \boldsymbol{u}_\eta &= 0, && (8) \end{aligned}$$

with appropriate external boundary conditions. Note that Eqs. (7)-(8) are valid in the *entire domain.* Here $\eta > 0$ is a penalization coefficient and χ_0 denotes the characteristic (or mask) function

$$\chi_0(\boldsymbol{x}, t) = \begin{cases} 1 & \text{if } \boldsymbol{x} \in O_i, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

As $\eta \to 0$, it was proved theoretically (Angot *et al.* 1999) that the solutions of the penalized equations (7)-(8) converge to that of the Navier–Stokes equations with the correct boundary conditions. More precisely, the upper bound on the global L^∞ error of the penalization was shown to be (Angot *et al.* 1999)

$$||\boldsymbol{u} - \boldsymbol{u}_\eta|| \leq C\eta^{1/4}. \tag{10}$$

In fact, we find that the actual error is slightly better, $O(\eta)$. It is important to note that η is an arbitrary parameter, independent of the spatial or temporal discretization, and thus the boundary conditions can be enforced to any desired accuracy by choosing η appropriately. This property distinguishes the Brinkman method from other penalization schemes and allows the error to be controlled precisely. We have found that $\eta = 10^{-4}$ gives drag curves correct to about 1%.

Due to the fact that the wavelet collocation method does not utilize divergence-free wavelets, we use a standard split-step method in time, where the first step produces a non-solenoidal velocity field. This intermediate velocity is then made divergence-free by solving a Poisson equation for the pressure. It is well-known that solving the Poisson equation is the most computationally intensive part of split step methods. Thus, in order to solve the Poisson equation efficiently, we developed a new wavelet collocation multilevel elliptic solver (Vasilyev & Kevlahan 2005).

Although it is a very flexible and simple method, Brinkman penalization does have two drawbacks. First, the large factor $1/\eta$ means that the term is stiff and must be solved implicitly. This is not difficult since the penalization term is simply a mask times the velocity. A stiffly stable time integration scheme for the penalized equations is used. The second drawback is that since the penalized equations are defined over the whole domain they must also be solved inside the obstacle. This adds some extra computational work, but since the flow penetrates to a skin depth of only $O(\eta^{1/2})$ (Kevlahan & Ghidaglia 2001) there are few grid points inside the obstacle. The sparseness of the grid inside the obstacle is clear in Fig. 4(b).

3 Stochastic Coherent Adaptive Large Eddy Simulation

The adaptive LES, hereafter referred to as Stochastic Coherent Adaptive Large Eddy Simulation (SCALES) methodology, is based on the premise that the most energetic coherent vortices (or structures) of a turbulent flow dominate mixing, heat transfer and other quantities of engineering interest, while the smaller incoherent background is only of interest because of how it effects the energetic coherent vortices (Goldstein & Vasilyev 2004). The SCALES equations, which describe the space-time evolution of the most energetic coherent eddies in a turbulent flow, can be formally obtained by applying the wavelet thresholding filter (5) to the Navier-Stokes equations. Disregarding the commutation error between wavelet-filtering and differentiation, the SCALES governing equations for incompressible flows are written as the following filtered continuity and momentum equations

$$\frac{\partial \overline{u_i}^{>\epsilon}}{\partial x_i} = 0\,, \tag{11}$$

$$\frac{\partial \overline{u_i}^{>\epsilon}}{\partial t} + \overline{u_j}^{>\epsilon}\frac{\partial \overline{u_i}^{>\epsilon}}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}^{>\epsilon}}{\partial x_i} + \nu\frac{\partial^2 \overline{u_i}^{>\epsilon}}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j}\,, \tag{12}$$

where ρ and ν are the constant density and kinematic viscosity of the fluid, while p stands for the pressure field. Like in the classical LES formulation, as a result of the filtering process, the unresolved quantities

$$\tau_{ij} = \overline{u_i u_j}^{>\epsilon} - \overline{u_i}^{>\epsilon}\,\overline{u_j}^{>\epsilon}\,, \tag{13}$$

commonly referred to as SGS stresses, are introduced. In this context, they can be thought of representing the effect of unresolved less energetic eddies on the dynamics of the resolved energetic coherent vortices. In order to close the filtered equation (12), a SGS model is required to express the unknown stresses (13) as a given function of the resolved velocity field. In practice, the isotropic part of the SGS stress tensor is usually incorporated by a modified filtered pressure variable, so that only the deviatoric part, hereafter noted with a star, $\tau^*_{ij} = \tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij}$, is actually modeled. Henceforth, the filtered momentum equation can be re-written as

$$\frac{\partial \overline{u_i}^{>\epsilon}}{\partial t} + \overline{u_j}^{>\epsilon} \frac{\partial \overline{u_i}^{>\epsilon}}{\partial x_j} = -\frac{\partial \overline{P}^{>\epsilon}}{\partial x_i} + \nu \frac{\partial^2 \overline{u_i}^{>\epsilon}}{\partial x_j \partial x_j} - \frac{\partial \tau^*_{ij}}{\partial x_j} \,, \tag{14}$$

where $\overline{P}^{>\epsilon} = \frac{\overline{p}^{>\epsilon}}{\rho} + \frac{1}{3}\tau_{kk}$.

It is worth stressing that for a suitably low value of the wavelet thresholding level ϵ, the resulting SGS field closely resembles Gaussian white noise and no modeling procedure is required in practice to recover low order DNS statistics. This approach, referred to as Coherent Vortex Simulation (CVS), which was originally introduced in (Farge *et al.* 1999), has been successfully applied to isotropic turbulence simulation and the results are discussed in Section 7.2.

Before reviewing the numerical implementation of the SCALES methodology, let us discuss in more details the wavelet filtering of the Navier-Stokes equations, in terms of both practical application and formal interpretation. Due to the one-to-one correspondence between wavelets and grid points, filtering each scalar field variable with the corresponding absolute scale would lead to numerical complications since each variable should be solved on a different numerical grid. In the present study, in order to avoid this difficulty, the coupled wavelet thresholding strategy is adopted. Namely, after constructing the mask of significant wavelet coefficients for each primary variable, the union of these masks results in a global thresholding mask that is used as a common mask for filtering all the variables. Moreover, according to the definition (5), the absolute filtering threshold should be theoretically based upon the values of the unfiltered variable, whereas, in a real SCALES calculation, the filtering procedure is actually based upon the values of the resolved filtered variable. However, as demonstrated in (Goldstein *et al.* 2005), this approximation is fully acceptable. For instance, regarding the velocity scale, in the homogeneous case one can use $U_i = \langle 2k_{\text{res}} \rangle^{1/2}$, where the angular brackets denote volume-averaging and

$$k_{\text{res}} = \frac{1}{2} \overline{u_j}^{>\epsilon} \, \overline{u_j}^{>\epsilon} \tag{15}$$

stands for the resolved kinetic energy.

As to wavelet filtering interpretation, one can view the wavelet thresholding procedure as a local spatially variable time-dependent low-pass filter that removes the high wavenumber components of the flow field. The local characteristic filter width, say $\Delta(\boldsymbol{x}, t)$, which is implicitly defined by the thresholding procedure and can be extracted from the global mask during the simulation, is to be interpreted as the actual turbulence-resolution length scale (Pope 2004). In fact, it is a measure of the local numerical resolution with the minimum allowable characteristic width corresponding to the highest level $j_{\max}$ in Eq.(5). The smaller the value of ϵ, the smaller the length scale Δ and the greater the fraction of resolved kinetic-energy in any local region of the domain. In the limit of vanishing ϵ the wavelet-based DNS solution is obtained over the whole domain. Such an interpretation of the wavelet thresholding filter highlights the similarity between the SCALES and the classical LES approaches. However, the wavelet filter is distinctively different from the usual filters adopted in LES, primarily because it changes in time following the flow evolution. That results in using a self-adaptive computational grid that tracks the areas of significant energy in the physical space during the simulation.

4 Global Dynamic SGS Model

The standard Smagorinsky (Smagorinsky 1963) eddy viscosity SGS stress model defines an eddy viscosity that is proportional to the filter width and the characteristic filtered rate of strain. In the case of the coupled nonlinear wavelet thresholding filter used in SCALES the filter width is implicitly defined by the non-dimensional wavelet threshold parameter, ϵ. Therefore in SCALES, ϵ is used to properly scale the eddy viscosity:

$$\nu_T = C_\epsilon \ell^2 \epsilon^\alpha \| \overline{S}^{>\epsilon} \| \, , \tag{16}$$

where C_ϵ is non-dimensional model coefficient, ℓ is the global characteristic length scale, and

$$\overline{S_{ij}}^{>\epsilon} = \frac{1}{2}\left(\frac{\partial \overline{u_i}^{>\epsilon}}{\partial x_j} + \frac{\partial \overline{u_j}^{>\epsilon}}{\partial x_i} \right) \tag{17}$$

is the strain rate of the resolved scales. Note that the model units do not depend on α simply because ϵ is non-dimensional. We will show in section 4.1 that appropriate scaling is obtained with $\alpha = 2$. The new linear eddy viscosity model is then used to define a model for the subgrid scale stress (13),

$$\overline{\tau^M_{ij}}^{>\epsilon} \equiv -2\nu_T \overline{S_{ij}}^{>\epsilon} \, , \tag{18}$$

where ν_T is the turbulent eddy viscosity.

The global characteristic length scale ℓ is introduced to obtain the proper units for the eddy viscosity ν_T. In addition, the length scale is assumed to be independent of the filter threshold parameter ϵ. With these two assumptions, the exact definition of ℓ does not need to be specified, since the whole group $C_\epsilon \ell^2 \epsilon^\alpha$ is determined by the dynamic procedure. ℓ^2 can be interpreted as an averaged characteristic length scale which is absorbed into the dynamic procedure. Currently we are working on the extension of the model that uses a local characteristic length scale ℓ, interpreted as the local characteristic vortical length scale implicitly defined by the wavelet thresholding filter.

The new Germano dynamic formulation for the model coefficient C_ϵ is, thus, based on the wavelet filter threshold parameter ϵ. For the dynamic procedure the grid filter is defined as $\overline{(\cdot)}^{>\epsilon}$ and the "test" filter is defined as $\overline{(\cdot)}^{>2\epsilon}$. The adjacent zone is excluded in both cases to obtain the proper model scaling. The dynamic procedure is then based on the original SGS stress equation (13), and an alternative subgrid scale stress,

$$\overline{T_{ij}}^{>2\epsilon} = \overline{\overline{u_i u_j}^{>\epsilon}}^{>2\epsilon} - \overline{\overline{u_i}^{>\epsilon}}^{>2\epsilon}\,\overline{\overline{u_j}^{>\epsilon}}^{>2\epsilon} \;, \tag{19}$$

which would result from applying the wavelet thresholding test filter ($\overline{(\cdot)}^{>2\epsilon}$) to (11–13). Note that the wavelet filter is a projection operator, so by definition

$$\overline{(\cdot)}^{>\epsilon_C} \equiv \overline{\overline{(\cdot)}^{>\epsilon_A}}^{>\epsilon_B} \;, \tag{20}$$

where $\epsilon_C = \max(\epsilon_A, \epsilon_B)$. Filtering (13) at the test filter level and subtracting it from (19) results in the modified Germano's identity (Germano *et al.* 1991)

$$\overline{T_{ij}}^{>2\epsilon} - \overline{\overline{\tau_{ij}}^{>\epsilon}}^{>2\epsilon} = \overline{\overline{u_i}^{>\epsilon}\,\overline{u_j}^{>\epsilon}}^{>2\epsilon} - \overline{\overline{u_i}^{>\epsilon}}^{>2\epsilon}\,\overline{\overline{u_j}^{>\epsilon}}^{>2\epsilon} \;. \tag{21}$$

Then, substituting the modeled SGS stresses at the two filter levels into (21) gives

$$\begin{aligned} \overline{T_{ij}}^{>2\epsilon} - \overline{\overline{\tau_{ij}}^{>\epsilon}}^{>2\epsilon} &\approx \overline{T^M_{ij}}^{>2\epsilon} - \overline{\overline{\tau^M_{ij}}^{>\epsilon}}^{>2\epsilon} \\ &= 2C_\epsilon \ell^2 (2\epsilon)^2 \|\overline{S}^{>2\epsilon}\| \overline{S_{ij}}^{>2\epsilon} - 2C_\epsilon \ell^2 \epsilon^2 \overline{\|\overline{S}^{>\epsilon}\| \overline{S_{ij}}^{>\epsilon}}^{>2\epsilon} \;. \end{aligned} \tag{22}$$

Following Lilly's (Lilly 1992) notation we define L_{ij} and M_{ij} as follows,

$$L_{ij} = \overline{\overline{u_i}^{>\epsilon}\,\overline{u_j}^{>\epsilon}}^{>2\epsilon} - \overline{\overline{u_i}^{>\epsilon}}^{>2\epsilon}\,\overline{\overline{u_j}^{>\epsilon}}^{>2\epsilon} \;, \tag{23}$$

$$M_{ij}\ell^2\epsilon^2 = 2\overline{\|\overline{S}^{>\epsilon}\| \overline{S_{ij}}^{>\epsilon}}^{>2\epsilon} - 8\|\overline{S}^{>2\epsilon}\| \overline{S_{ij}}^{>2\epsilon} \;, \tag{24}$$

where L_{ij} is the wavelet-filtered analog of the Leonard stress. This results in an overdetermined system of equations that can be used to find $C_\epsilon \ell^2 \epsilon^2$

$$C_\epsilon \ell^2 \epsilon^2 M_{ij} = L_{ij} \;. \tag{25}$$

Following Lilly's (Lilly 1992) least square solution to this system, we obtain the following expression for the local Smagorinsky model coefficient,

$$C_\epsilon \ell^2 \epsilon^2 = \frac{L_{ij} M_{ij}}{M_{ij} M_{ij}} \;. \tag{26}$$

With this model formulation $C_\epsilon \ell^2 \epsilon^2$ can be locally positive or negative, which allows for local backscatter of energy to resolved scales. In practice it has been found that locally negative values of $C_\epsilon \ell^2 \epsilon^2$ cause numerical instabilities in SCALES, as in LES, so we average over homogeneous directions:

$$C_\epsilon \ell^2 \epsilon^2 = \frac{\langle L_{ij} M_{ij} \rangle}{\langle M_{ij} M_{ij} \rangle} \;, \tag{27}$$

where $\langle \cdot \rangle$ denotes volume averaging.

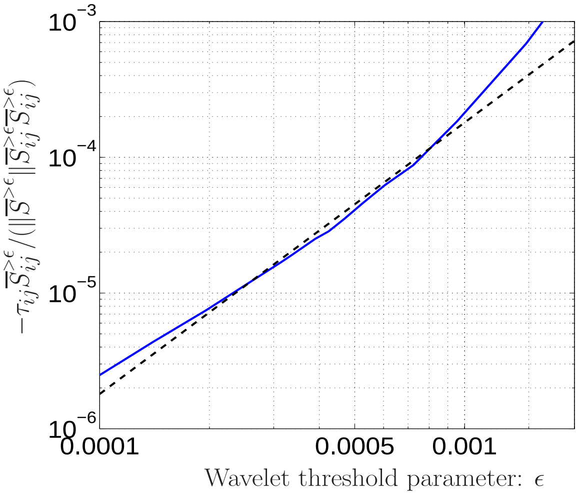


Figure 5: $\tau_{ij}\overline{S}_{ij}^{>\epsilon}/(\|\overline{S}^{>\epsilon}\|\overline{S}_{ij}^{>\epsilon}\overline{S}_{ij}^{>\epsilon})$ vs. relative wavelet threshold parameter ϵ using velocity wavelet filtering, without adjacent zone (———) for field F_{256}. The dashed line is ϵ^2. The L_∞ WTF-norm was used for this *a priori* test. This range of ϵ corresponds to a field compression over the range of 78.5% to 99.95%. It can be seen that $\tau_{ij}\overline{S}_{ij}^{>\epsilon}/(\|\overline{S}^{>\epsilon}\|\overline{S}_{ij}^{>\epsilon}\overline{S}_{ij}^{>\epsilon})$ scales roughly as ϵ^2. The scaling begins to deviate at $\epsilon = 0.001$, which corresponds to 99.4% compression.

4.1 Model Scaling

If we make the assumption that, with an appropriate value for α, the eddy viscosity model (16–18) provides the right dissipation it is easy to show,

$$2C_\epsilon \ell^2 \epsilon^\alpha = -\frac{\left\langle \tau_{ij}\overline{S}_{ij}^{>\epsilon}\right\rangle}{\left\langle \|\overline{S}^{>\epsilon}\|\overline{S}_{ij}^{>\epsilon}\overline{S}_{ij}^{>\epsilon}\right\rangle}\,, \tag{28}$$

where α is the scaling law and ℓ is taken to be constant over the domain. The correct scaling is determined from *a priori* testing, using the isotropic turbulence field F_{256}. In Fig. 5 the scaling of $-\left\langle \tau_{ij}\overline{S}_{ij}^{>\epsilon}\right\rangle / \left\langle \|\overline{S}^{>\epsilon}\|\overline{S}_{ij}^{>\epsilon}\overline{S}_{ij}^{>\epsilon}\right\rangle$ is shown over a range of ϵ that corresponds to a field compression over the range of 78.5% to 99.95%. The slope of the curve in log-log axis determines the appropriate ϵ scaling. As can be seen, the quantity $\left\langle \tau_{ij}\overline{S}_{ij}^{>\epsilon}\right\rangle / \left\langle \|\overline{S}^{>\epsilon}\|\overline{S}_{ij}^{>\epsilon}\overline{S}_{ij}^{>\epsilon}\right\rangle$ scales roughly as ϵ^2 for a wide range of compressions. However some deviation from this scaling is observed above 99.4% compression. Based on this *a priori* test of the scaling, the new dynamic Smagorinsky-type eddy viscosity model (16) has been implemented. The results of simulations with this new SGS model are discussed in section 7.

5 Local Dynamic SGS Models

The global dynamic eddy viscosity model described in previous section has a major drawback, namely the use of a global (spatially non-variable) Smagorinsky model coefficient. The use of a global dynamic model unnecessarily limits the SCALES approach to flows with at least one homogeneous direction. This is unfortunate since the dynamic adaptability of SCALES is ideally suited to fully inhomogeneous flows. In order to realize the full benefits of SCALES in highly non-homogenous flows in complex geometries a local SGS model is required. In this section two different formulations of local dynamic SGS model that were explored are reported:

1. Modified Germano's dynamic procedure redefined in terms of wavelet thresholding filters with a modified Lagrangian path-line/tube averaging procedure (published in (Vasilyev *et al.* 2008)), and
2. SGS kinetic energy based model, where an additional transport equation for the SGS kinetic energy is solved to enforce the energy budget between resolved and unresolved motions (published in (De Stefano *et al.* 2008)).

5.1 Lagrangian Dynamic Local SGS Model

As mentioned in previous section, the local dynamic model (26) cause numerical instabilities in SCALES, as in LES, so the averaging procedure is required to stabilize the model. One way to achieve this is to average over homogeneous directions, which results in global (spatially non-variable) dynamic model (27). This limits the applicability of the model to flows with at least one homogeneous direction. Another way to stabilize the model is to perform Lagrangian pathline averaging procedure originally introduced in (Meneveau *et al.* 1996). To start the SGS stress tensor (18) is rewritten in the following form

$$\tau^*_{ij} \cong -2C_S\Delta^2\epsilon^2\left|\overline{S}^{>\epsilon}\right|\overline{S_{ij}}^{>\epsilon}, \tag{29}$$

where $\overline{S_{ij}}^{>\epsilon} = \frac{1}{2}\left(\frac{\partial\overline{u_i}^{>\epsilon}}{\partial x_j} + \frac{\partial\overline{u_j}^{>\epsilon}}{\partial x_i}\right)$ is the resolved rate-of-strain tensor and $\Delta(\mathbf{x},t)$ is the *local* characteristic filter lengthscale *dynamically* defined by the wavelet thresholding filter. Note that Δ is distinctively different from the classical LES, where the local filter width is defined *statically.* Also note that the variable filter width, Δ, is the key-parameter in the SCALES formulation as it strictly reflects the adaptive nature of the method. Once a wavelet threshold is given, the corresponding thresholding mask implicitly defines a point-wise time-dependent filter width. This is different from classical LES where the local, possibly non-uniform, filter width is defined *a priori* and does not depend on the actual flow realization.

Following the modified Germano's dynamic procedure redefined in terms of two wavelet thresholding filters, as discussed in Section 4, the SGS stress corresponding to the wavelet test filter at twice the threshold, noted $\overline{(\cdot)}^{>2\epsilon}$, is defined as

$$T_{ij} = \overline{u_i u_j}^{>2\epsilon} - \overline{u_i}^{>2\epsilon}\,\overline{u_j}^{>2\epsilon}. \tag{30}$$

Note that, the wavelet filter being a projection operator, by definition, it holds $\overline{\overline{(\cdot)}^{>\epsilon}}^{>2\epsilon} \equiv \overline{(\cdot)}^{>2\epsilon}$. Filtering Eq. (13) at the test filter level and combining with Eq. (59) results in the following modified Germano identity for the Leonard stresses:

$$L_{ij} \equiv T_{ij} - \overline{\tau_{ij}}^{>2\epsilon} = \overline{\overline{u_i}^{>\epsilon}\overline{u_j}^{>\epsilon}}^{>2\epsilon} - \overline{u_i}^{>2\epsilon}\,\overline{u_j}^{>2\epsilon}. \tag{31}$$

Exploiting the model (29) and the analogous relation for the test filtered SGS stresses

$$T^*_{ij} \cong -2C_S\Delta^2\left(2\epsilon\right)^2\left|\overline{S}^{>2\epsilon}\right|\overline{S_{ij}}^{>2\epsilon}, \tag{32}$$

one obtains

$$2\overline{C_S\Delta^2\epsilon^2\left|\overline{S}^{>\epsilon}\right|\overline{S_{ij}}^{>\epsilon}}^{>2\epsilon} - 2C_S\Delta^2\left(2\epsilon\right)^2\left|\overline{S}^{>2\epsilon}\right|\overline{S_{ij}}^{>2\epsilon} = L^*_{ij}. \tag{33}$$

A least square solution to (Eq. 33) leads to the following local Smagorinsky model coefficient definition:

$$C_S(\mathbf{x},t)\epsilon^2 = \frac{L^*_{ij}M_{ij}}{M_{nk}M_{nk}}, \tag{34}$$

where

$$M_{ij} \equiv 2\Delta^2\left[\overline{\left|\overline{S}^{>\epsilon}\right|\overline{S_{ij}}^{>\epsilon}}^{>2\epsilon} - 4\left|\overline{S}^{>2\epsilon}\right|\overline{S_{ij}}^{>2\epsilon}\right]. \tag{35}$$

The coefficient C_S can be actually positive or negative, which allows for local backscatter of energy from unresolved to resolved modes. However, it has been found that negative values of C_S cause numerical instabilities. To avoid this, for homogeneous flow, one can introduce an average over homogeneous directions. This procedure results in a global dynamic model analogous to the one discussed in the previous section.

In this study we follow a Lagrangian dynamic model formulation (Meneveau *et al.* 1996) and take the statistical averages over the trajectory of a fluid particle:

$$\mathcal{I}_{LM}\left(\boldsymbol{x},t\right) = \frac{1}{T}\int_{-\infty}^{t} e^{\frac{\tau-t}{T}} L_{ij}\left(\boldsymbol{x}\left(\tau\right),\tau\right)M_{ij}\left(\boldsymbol{x}\left(\tau\right),\tau\right)d\tau, \tag{36}$$

$$\mathcal{I}_{MM}\left(\boldsymbol{x},t\right) = \frac{1}{T}\int_{-\infty}^{t} e^{\frac{\tau-t}{T}} M_{ij}\left(\boldsymbol{x}\left(\tau\right),\tau\right)M_{ij}\left(\boldsymbol{x}\left(\tau\right),\tau\right)d\tau, \tag{37}$$

which leads to the following local Smagorinsky model coefficient:

$$C_S(\mathbf{x},t)\epsilon^2 = \frac{\mathcal{I}_{LM}}{\mathcal{I}_{MM}}. \tag{38}$$

To avoid the computationally expensive procedure of Lagrangian path-line averaging, we follow (Meneveau *et al.* 1996) and differentiate Eqs. (36) and (37) with respect to time to obtain the following evolution equations for $\mathcal{I}_{LM}$ and $\mathcal{I}_{MM}$:

$$\frac{\partial\mathcal{I}_{LM}}{\partial t} + \overline{u_l}^{>\epsilon}\frac{\partial\mathcal{I}_{LM}}{\partial x_l} = \frac{1}{T}(L_{ij}M_{ij} - \mathcal{I}_{LM}), \tag{39}$$

$$\frac{\partial\mathcal{I}_{MM}}{\partial t} + \overline{u_l}^{>\epsilon}\frac{\partial\mathcal{I}_{MM}}{\partial x_l} = \frac{1}{T}(M_{nk}M_{nk} - \mathcal{I}_{MM}). \tag{40}$$

As in (Meneveau *et al.* 1996) the relaxation time scale T is defined as $T(\mathbf{x},t) = \theta\Delta\ (\mathcal{I}_{LM}\mathcal{I}_{MM})^{-1/8}$, θ being a dimensionless parameter of order unity.

The equations (39) and (40) should be solved together with the SCALES equations, (11) and (12). It should be noticed that both $\mathcal{I}_{LM}$ and $\mathcal{I}_{MM}$ have higher frequency content when compared to the velocity field. This is due to two main factors: the quartic character of nonlinearity of $\mathcal{I}_{LM}$ and $\mathcal{I}_{MM}$ with respect to velocity and the creation of small scales due to chaotic convective mixing. Thus, in order to adequately resolve both $\mathcal{I}_{LM}$ and $\mathcal{I}_{MM}$, one needs to have a substantially finer computational mesh than the one required by the velocity field, which is impractical. To by-pass this problem we consider two different extensions of the Lagrangian path-line averaging: Lagrangian path-tube averaging and Lagrangian path-line diffusive averaging.

The Lagrangian path-tube averaging consist of taking the statistical filtered averages over the trajectory of a fluid particle:

$$\mathcal{I}_{LM}\left(\boldsymbol{x},t\right) = \frac{1}{T}\int_{-\infty}^{t}\iiint_{D} e^{\frac{\tau-t}{T}} G\left(\boldsymbol{y}-\boldsymbol{x}\left(\tau\right),\ \boldsymbol{x}\left(\tau\right)\right) L_{ij}\left(\boldsymbol{y},\tau\right) M_{ij}\left(\boldsymbol{y},\tau\right) d\tau d\boldsymbol{y}, \tag{41}$$

$$\mathcal{I}_{MM}\left(\boldsymbol{x},t\right) = \frac{1}{T}\int_{-\infty}^{t}\iiint_{D} e^{\frac{\tau-t}{T}} G\left(\boldsymbol{y}-\boldsymbol{x}\left(\tau\right),\ \boldsymbol{x}\left(\tau\right)\right) M_{hk}\left(\boldsymbol{y},\tau\right) M_{hk}\left(\boldsymbol{y},\tau\right) d\tau d\boldsymbol{y}, \tag{42}$$

where $G\left(\boldsymbol{\xi},\ \boldsymbol{x}\right)$ is the local, location dependent, low-pass filter with the second variable denoting filter location. Note that the low-pass filter averages the values in the neighborhood of the path-line, effectively making it path-tube averaging. Also note that if $G\left(\boldsymbol{\xi},\boldsymbol{x}\right) = 1$, *i.e.* no additional spatial filter is applied, the formulations are identical to the one in (Meneveau *et al.* 1996). This modified averaging procedure leads to the following evolution equations for the auxiliary variables $\mathcal{I}_{LM}$ and $\mathcal{I}_{MM}$:

$$\frac{\partial\mathcal{I}_{LM}}{\partial t} + \overline{u}_l^{>\epsilon}\frac{\partial\mathcal{I}_{LM}}{\partial x_l} = \frac{1}{T}\left(\ \overline{L_{ij}M_{ij}}^{\text{LP}} - \mathcal{I}_{LM}\right), \tag{43}$$

$$\frac{\partial\mathcal{I}_{MM}}{\partial t} + \overline{u}_l^{>\epsilon}\frac{\partial\mathcal{I}_{MM}}{\partial x_l} = \frac{1}{T}\left(\ \overline{M_{nk}M_{nk}}^{\text{LP}} - \mathcal{I}_{MM}\right), \tag{44}$$

where $\overline{(\cdot)}^{\text{LP}}$ defines low-pass filtering based on $G\left(\boldsymbol{x},\tau\right)$.

In the Lagrangian path-line diffusive averaging an additional artificial diffusion term is added to the evolution equations:

$$\frac{\partial\mathcal{I}_{LM}}{\partial t} + \overline{u}_l^{>\epsilon}\frac{\partial\mathcal{I}_{LM}}{\partial x_l} = \frac{1}{T}(L_{ij}M_{ij} - \mathcal{I}_{LM}) + \mathcal{D}_{\mathcal{I}}\frac{\partial^2\mathcal{I}_{LM}}{\partial x_j\partial x_j}, \tag{45}$$

$$\frac{\partial\mathcal{I}_{MM}}{\partial t} + \overline{u}_l^{>\epsilon}\frac{\partial\mathcal{I}_{MM}}{\partial x_l} = \frac{1}{T}(M_{nk}M_{nk} - \mathcal{I}_{MM}) + \mathcal{D}_{\mathcal{I}}\frac{\partial^2\mathcal{I}_{MM}}{\partial x_j\partial x_j}. \tag{46}$$

To avoid the creation of small scales, the diffusion time scale, $\Delta^2/\mathcal{D}_{\mathcal{I}}$, should be smaller than the convective time scale associated with local strain, $\left|\overline{S}^{>\epsilon}\right|^{-1}$, which results in $\mathcal{D}_{\mathcal{I}} = C_{\mathcal{I}}\Delta_\epsilon^2\left|\overline{S}^{>\epsilon}\right|$, where C_I is a dimensionless parameter of order unity. Note that the Lagrangian path-line diffusive averaging procedure can be formally derived by using differential implicit filters as described in (Vasilyev *et al.* 2008).

Combining these two approaches results in the Lagrangian path-line/tube averaged equations:

$$\frac{\partial \mathcal{I}_{LM}}{\partial t} + \overline{u}_l^{>\epsilon}\frac{\partial \mathcal{I}_{LM}}{\partial x_l} = \frac{1}{T}\left(\overline{L_{ij}M_{ij}}^{\,\mathrm{LP}} - \mathcal{I}_{LM}\right) + \mathcal{D}_{\mathcal{I}}\frac{\partial^2 \mathcal{I}_{LM}}{\partial x_j \partial x_j}, \tag{47}$$

$$\frac{\partial \mathcal{I}_{MM}}{\partial t} + \overline{u}_l^{>\epsilon}\frac{\partial \mathcal{I}_{MM}}{\partial x_l} = \frac{1}{T}\left(\overline{M_{nk}M_{nk}}^{\,\mathrm{LP}} - \mathcal{I}_{MM}\right) + \mathcal{D}_{\mathcal{I}}\frac{\partial^2 \mathcal{I}_{MM}}{\partial x_j \partial x_j}. \tag{48}$$

Note that the case $G = 1$ and $C_{\mathcal{I}} = 0$ is equivalent to the original Lagrangian formulation of (Meneveau *et al.* 1996).

5.2 Kinetic Energy Based Modeling

An alternative mechanism to achieve locality of the SGS model is to solve an additional transport equation for the subgrid scale kinetic energy, k_{sgs}.

In order to take full advantage of the SCALES methodology for simulating complex turbulent flows, the development of localized closure models appears necessary. For this purpose, a modeling mechanism that takes into account the local kinetic energy transfer back and forth between resolved and unresolved eddies can be exploited. In fact, as the closure model is mainly required to provide the right rate of energy dissipation from the resolved field, the model coefficient can be calibrated upon the energy level of the residual motions. It has been demonstrated in (Ghosal *et al.* 1995) that energy-based localized models for LES can be successfully constructed by incorporating a transport model equation for the residual kinetic energy. Moreover, the use of the kinetic energy variable appears as a natural choice in the present context, given the main feature of the SCALES approach, which consists in solving for the significant part of the energy content of the flow-field, while modeling the effect of the less-energetic background flow. In this work the use of both local eddy-viscosity and non-eddy viscosity kinetic energy-based models in the context of SCALES is explored.

In order to address some issues about the local energy transfer between resolved and residual motions, let us first consider the balance equation for the resolved kinetic energy, *i.e.* , according to Eq.(12),

$$\frac{\partial k_{\mathrm{res}}}{\partial t} + \overline{u}_j^{>\epsilon}\frac{\partial k_{\mathrm{res}}}{\partial x_j} = -\frac{\partial}{\partial x_i}\left[\overline{u_j}^{>\epsilon}\left(\tau^*_{ij} + \overline{P}^{>\epsilon}\delta_{ij}\right)\right] + \nu\frac{\partial^2 k_{\mathrm{res}}}{\partial x_j \partial x_j} - \epsilon_{\mathrm{res}} - \Pi\,, \tag{49}$$

where $\epsilon_{\mathrm{res}} = \nu\frac{\partial \overline{u_i}^{>\epsilon}}{\partial x_j}\frac{\partial \overline{u_i}^{>\epsilon}}{\partial x_j}$ stands for the rate of resolved viscous dissipation and $\Pi = -\tau^*_{ij}\overline{S}_{ij}^{>\epsilon}$ represents the rate at which energy is transferred to unresolved residual motions. As to resolved viscous dissipation, it is worth pointing out that $\epsilon_{\rm res}$ is not negligible for SCALES, in contrast to what typically happens for classical LES. This is mostly due to the adaptive nature of the SCALES approach, which results in the presence of significant energy at small scales, as demonstrated in the Sections 7.4, 7.5, and 7.6. Note also that the local energy transfer Π can show both signs, even though energy is globally transferred from resolved to residual motions, *e.g.*, $\langle \Pi \rangle > 0$ for isotropic turbulence. For this reason, Π is commonly referred to as SGS dissipation.

The subgrid-scale kinetic energy, say $k_{\rm sgs}$, is formally defined as the difference between the wavelet filtered energy and the kinetic energy of the filtered velocity field, $k_{\rm res}$, that is

$$k_{\rm sgs} = \frac{1}{2}\left(\overline{u_i u_i}^{>\epsilon} - \overline{u_i}^{>\epsilon}\overline{u_i}^{>\epsilon}\right) . \tag{50}$$

The above energy variable is simply related to the trace of the SGS stress tensor, being $k_{\rm sgs} = \frac{1}{2}\tau_{ii}$. Note that the adopted terminology is in some way inappropriate as this quantity does not stand for the kinetic energy associated with the SGS motions, which is $\frac{1}{2}\overline{u_i' u_i'}^{>\epsilon}$, $u_i' = u_i - \overline{u_i}^{>\epsilon}$ being the residual velocity field. The evolution of $k_{\rm sgs}$ can be modeled by means of the following transport equation (*e.g.*, (Ghosal *et al.* 1995))

$$\frac{\partial k_{\rm sgs}}{\partial t} + \overline{u}_j^{>\epsilon}\frac{\partial k_{\rm sgs}}{\partial x_j} = \nu \frac{\partial^2 k_{\rm sgs}}{\partial x_j \partial x_j} - \epsilon_{\rm sgs} + \Pi \, , \tag{51}$$

where $\epsilon_{\rm sgs}$ stands for the viscous dissipation rate of the SGS kinetic energy that is the unclosed term

$$\epsilon_{\rm sgs} = \nu \left(\overline{\frac{\partial u_i}{\partial x_j}\frac{\partial u_i}{\partial x_j}}^{>\epsilon} - \frac{\partial \overline{u_i}^{>\epsilon}}{\partial x_j}\frac{\partial \overline{u_i}^{>\epsilon}}{\partial x_j} \right) . \tag{52}$$

In order to close the energy equation (51), a further model for the SGS viscous dissipation $\epsilon_{\rm sgs}$ must be introduced, as discussed in the following section.

The SGS energy production Π takes a fundamental role in modeling procedures based on the kinetic energy variable. As it contributes with different sign to both resolved (49) and SGS (51) energy balance, it can be exploited to develop a built-in feedback mechanism that automatically stabilizes the numerical solution. This way, no averaging procedure is needed and the full locality of the model is achieved. Namely, one can assume the SGS dissipation to be a monotonic increasing function of k_{sgs} so that, for example, if there is energy backscatter from unresolved to resolved motions (*i.e.* , $\Pi < 0$) the resolved kinetic energy locally increases while the residual one decreases, but the SGS forcing decreases as well so leading to the suppression of the reverse flow of energy.

5.3 Local dynamic energy-based eddy-viscosity models

The first step in building localized SGS models is taken by considering eddy-viscosity models where the turbulent viscosity no longer depends upon the resolved rate-of-strain, as in the

Smagorinsky approach, but on the SGS kinetic energy. In eddy viscosity based models the unknown SGS stress tensor in Eq.(12) is approximated by

$$\tau^*_{ij} \cong -2\nu_t \overline{S_{ij}}^{>\epsilon} \ , \tag{53}$$

where $\overline{S_{ij}}^{>\epsilon} = \frac{1}{2}\left(\frac{\partial \overline{u_i}^{>\epsilon}}{\partial x_j} + \frac{\partial \overline{u_j}^{>\epsilon}}{\partial x_i}\right)$ is the resolved rate-of-strain tensor and ν_t stands for the turbulent viscosity, which is the model parameter to be expressed in terms of the resolved field. Similarly to what done in (Ghosal *et al.* 1995), let us take the square root of k_{sgs} as the velocity scale and the wavelet-filter characteristic width Δ as the length scale for the turbulent eddy-viscosity definition that is

$$\nu_t = C_\nu \Delta k_{\mathrm{sgs}}^{1/2} \ , \tag{54}$$

C_ν being the dimensionless coefficient to be determined. This way, Eq.(53) is rewritten as

$$\tau^*_{ij} \cong -2C_\nu \Delta k_{\mathrm{sgs}}^{1/2} \overline{S}_{ij}^{>\epsilon} \tag{55}$$

and the SGS dissipation rate is approximated in terms of the SGS kinetic energy as

$$\Pi \cong C_\nu \Delta k_{\mathrm{sgs}}^{1/2} |\overline{S}^{>\epsilon}| \tag{56}$$

where $|\overline{S}^{>\epsilon}| = 2\overline{S}_{ij}^{>\epsilon}\overline{S}_{ij}^{>\epsilon}$. Note that the SGS dissipation rate can show both signs, thus allowing for the simulation of local energy backscatter.

Given the eddy-viscosity nature of the model, when solving for the SGS energy, the additional diffusion due to the turbulent viscosity is considered, so that the energy equation (51) is rewritten as

$$\frac{\partial k_{\mathrm{sgs}}}{\partial t} + \overline{u}_j^{>\epsilon}\frac{\partial k_{\mathrm{sgs}}}{\partial x_j} = (\nu + \nu_t)\frac{\partial^2 k_{\mathrm{sgs}}}{\partial x_j \partial x_j} - \epsilon_{\mathrm{sgs}} + \Pi \ . \tag{57}$$

As mentioned above, in addition to the SGS stress model, the SGS energy dissipation model for ϵ_{sgs} is needed. The latter variable can be modeled, using simple scaling arguments, as

$$\epsilon_{\mathrm{sgs}} = C_\epsilon \frac{k_{\mathrm{sgs}}^{3/2}}{\Delta} \ , \tag{58}$$

C_ϵ being the second dimensionless model coefficient (*e.g.*, (Schumann 1975, Ghosal *et al.* 1995)) to be determined. Another possibility, not taken here, would be to consider an additional evolution model equation for ϵ_{sgs}, as done for instance in (Pomraning & Rutland 2002).

The wavelet-filtered Navier-Stokes equations (12) and the SGS kinetic energy equation (57) stand for a closed system of coupled equations that is solved with the AWCM methodology briefly described in Section 2.5. In particular, the global thresholding mask for wavelet filtering can be constructed by considering both the velocity and the SGS kinetic energy fields.

In a first lighter version of the model, in order to save computational resources, the model parameters C_ν and C_ϵ are *a-priori* prescribed. In particular, the unit value for C_ϵ is fixed, as typically done in LES based on a similar approach. Also, the empirical value $C_\nu = 0.06$ is prescribed for the turbulent viscosity coefficient, as a result of acceptable global matching with the wavelet-filtered DNS solution for the numerical experiments carried out in this work. This one-equation model will be referred to as the localized kinetic energy-based model (for following discussion, LKM). It is worth stressing that, though the model coefficients are fixed, the LKM procedure is nevertheless "dynamic" in some way as it implicitly takes into account the local energy transfer between the resolved and unresolved motions for the ongoing simulation.

A fully dynamic version of the kinetic energy-based eddy-viscosity model, with the model coefficients not prescribed but derived from the actual resolved field using the classical Germano dynamic approach (Germano *et al.* 1991), is developed as illustrated in the following. Let us introduce a secondary test-filter with a characteristic filter width $\widehat{\Delta} > \Delta$, formally denoting the test-filtered resolved velocity as $\widehat{\overline{u_i}^{>\epsilon}}$. The stress tensor at the test level is given by

$$T_{ij} = \widehat{\overline{u_i u_j}^{>\epsilon}} - \widehat{\overline{u_i}^{>\epsilon}}\,\widehat{\overline{u_j}^{>\epsilon}} \ , \tag{59}$$

so that, filtering (13) at the test level and combining with (59), results in the following definition for the Leonard stresses

$$L_{ij} = \widehat{\overline{u_i}^{>\epsilon}\overline{u_j}^{>\epsilon}} - \widehat{\overline{u_i}^{>\epsilon}}\,\widehat{\overline{u_j}^{>\epsilon}} \ , \tag{60}$$

or, equivalently, the popular Germano identity

$$T_{ij} - \widehat{\tau}_{ij} = L_{ij} \ . \tag{61}$$

Once the test-filter is given, the Leonard stresses are directly computable upon the resolved velocity field and can be exploited to determine the model coefficient with no *a-priori* prescriptions. Differently from what is done in (Goldstein *et al.* 2005), here a low-pass discrete filter is used. Specifically, the discrete low-pass test-filter is constructed using the adjacent grid-points ensuring the proper filter width and positivity of the filter weights.

The unresolved kinetic energy at the test level, which is referred to as the subtest-scale (STS) kinetic energy is defined as

$$k_{\rm sts} = \widehat{\overline{u_i u_i}^{>\epsilon}} - \widehat{\overline{u_i}^{>\epsilon}}\widehat{\overline{u_i}^{>\epsilon}} \ , \tag{62}$$

that is $k_{\rm sts} = \frac{1}{2}T_{ii}$. By analogy with (51), the transport model equation for $k_{\rm sts}$ can be written as

$$\frac{\partial k_{\rm sts}}{\partial t} + \widehat{\overline{u_j}^{>\epsilon}}\frac{\partial k_{\rm sts}}{\partial x_j} = (\nu + \nu_t)\frac{\partial^2 k_{\rm sts}}{\partial x_j \partial x_j} - \epsilon_{\rm sts} + \Pi_{\rm sts} \ , \tag{63}$$

where the STS energy viscous dissipation rate is

$$\epsilon_{\text{sts}} = \nu \left(\widehat{\overline{\frac{\partial u_i}{\partial x_j}\frac{\partial u_i}{\partial x_j}}^{>\epsilon}} - \frac{\partial \widehat{\overline{u_i}^{>\epsilon}}}{\partial x_j}\frac{\partial \widehat{\overline{u_i}^{>\epsilon}}}{\partial x_j} \right) \tag{64}$$

and Π_{sts} stands for the STS energy production that is $-T^*_{ij}\widehat{\overline{S_{ij}}^{>\epsilon}}$, $\widehat{\overline{S_{ij}}^{>\epsilon}} = \frac{1}{2}\left(\frac{\partial \widehat{\overline{u_i}^{>\epsilon}}}{\partial x_j} + \frac{\partial \widehat{\overline{u_j}^{>\epsilon}}}{\partial x_i}\right)$ being the resolved rate-of-strain tensor at the test level. In a similar manner, the kinetic energy that is resolved at the test level can be defined as

$$k_{\text{rts}} = \widehat{\overline{u_i}^{>\epsilon}\overline{u_i}^{>\epsilon}} - \widehat{\overline{u_i}^{>\epsilon}}\widehat{\overline{u_i}^{>\epsilon}}\ , \tag{65}$$

or, equivalently, $k_{\text{rts}} = \frac{1}{2}L_{ii}$, owing to the Leonard stress definition (60). It is worth noting that, thanks to the positiveness of the employed test-filter, the variable k_{rts} is always non-negative in the flow-field. This way, the Germano identity (61) can be re-written in terms of the kinetic energy variable as it follows

$$k_{\text{sts}} - \widehat{k_{\text{sgs}}} = k_{\text{rts}}\ . \tag{66}$$

The above relation allows for the STS kinetic energy to be directly expressed in terms of resolved quantities, which are the velocity and the SGS energy fields. After defining the resolved viscous dissipation at the test level

$$\epsilon_{\text{rts}} = \nu \left(\widehat{\frac{\partial \overline{u_i}^{>\epsilon}}{\partial x_j}\frac{\partial \overline{u_i}^{>\epsilon}}{\partial x_j}} - \frac{\partial \widehat{\overline{u_i}^{>\epsilon}}}{\partial x_j}\frac{\partial \widehat{\overline{u_i}^{>\epsilon}}}{\partial x_j} \right)\ , \tag{67}$$

a similar Germano identity relates the energy dissipation rates at test and grid levels

$$\epsilon_{\text{sts}} - \widehat{\epsilon_{\text{sgs}}} = \epsilon_{\text{rts}}\ . \tag{68}$$

Again, due to the positiveness of the test-filter, the variable ϵ_{rts} is always non-negative in the flow-field. It is worth pointing out that the identity (68) is actually unusable for classical LES formulations, since the scale separation acts in the inertial range and the resolved LES field does not contain significant contribution from dissipative scales (Ghosal *et al.* 1995). Conversely, equation (68) can be successfully exploited in the SCALES approach, where also the small-scale energetic structures are resolved.

Different fully dynamic versions of the energy-based eddy-viscosity modeling procedure are presented in the following sections. They are based on either a Bardina-like or a Germano-like approximation for the dynamic determination of the model coefficients C_ν and C_ϵ as space-time functions.

5.3.1 Eddy-viscosity modeling

Two different dynamic procedures are proposed to determine the unknown model coefficient for the turbulent eddy-viscosity (54).

Bardina-like model By analogy with (55), let us assume the Leonard stress can be approximated in terms of the resolved test-scale kinetic energy as follows

$$L_{ij}^{*} \cong -2C_\nu \widehat{\Delta} k_{\mathrm{rts}}^{1/2} \widehat{\overline{S}_{ij}^{>\epsilon}} \, , \tag{69}$$

where, as usual, the star denotes the deviatoric part. The above expression represents a system of five independent equations with the unique unknown C_ν, which can be approximately solved by exploiting a least-square methodology. That leads to

$$2C_\nu(\boldsymbol{x},t) = \frac{L_{ij}^{*}\sigma_{ij}}{\sigma_{ln}\sigma_{ln}} \, , \tag{70}$$

where, for simplifying the notation, the known tensor $\sigma_{ij} = -\alpha\Delta k_{\mathrm{rts}}^{1/2} \widehat{\overline{S}_{ij}^{>\epsilon}}$ is defined (Kim & Menon 1999), $\alpha = \widehat{\Delta}/\Delta$ being the test filter to grid ratio.

Germano-like model As an analog of (55), let us assume the STS stress can be approximated in terms of the STS kinetic energy as follows

$$T_{ij}^{*} \cong -2C_\nu \widehat{\Delta} k_{\mathrm{sts}}^{1/2} \widehat{\overline{S}_{ij}^{>\epsilon}} \, . \tag{71}$$

Therefore, according to the Germano identity (61), combining (55) and (71), it holds that

$$-2C_\nu \widehat{\Delta} k_{\mathrm{sts}}^{1/2} \widehat{\overline{S}_{ij}^{>\epsilon}} + C_\nu \Delta \widehat{k_{\mathrm{sgs}}^{1/2} \overline{S}_{ij}^{>\epsilon}} = L_{ij}^{*} \, , \tag{72}$$

where the coefficient C_ν is assumed to vary slowly in space so that it can be taken out of the test filtering operation. By exploiting the identity (66) and defining

$$M_{ij} = \widehat{k_{\mathrm{sgs}}^{1/2} \overline{S}_{ij}^{>\epsilon}} - \alpha(k_{\mathrm{rts}} + \widehat{k_{\mathrm{sgs}}})^{1/2} \widehat{\overline{S}_{ij}^{>\epsilon}} \, , \tag{73}$$

a least square solution to (72) leads to the determination of the following local model coefficient

$$2C_\nu(\boldsymbol{x},t)\Delta = \frac{L_{ij}^{*} M_{ij}}{M_{ln}M_{ln}} \, . \tag{74}$$

It is worth stressing that both the present modeling procedures, though based on the same SGS energy-based eddy-viscosity concept, nevertheless are very different from the dynamic localization model proposed in (Ghosal *et al.* 1995), where the model coefficient was determined by solving an integral equation in the framework of a constrained variational problem.

5.3.2 SGS energy dissipation modeling

Two different dynamic procedures are proposed to determine the unknown model coefficient for the SGS energy dissipation (58).

Bardina-like model According to a Bardina-like approach, by analogy with (58), let us assume the resolved test-scale energy dissipation can be approximated as

$$\epsilon_{\rm rts} \cong C_\epsilon \frac{k_{\rm rts}^{3/2}}{\widehat{\Delta}} \, . \tag{75}$$

The above equation can be solved for the unknown C_ϵ resulting in the following local model coefficient

$$\frac{C_\epsilon(\boldsymbol{x},t)}{\Delta} = \frac{\alpha\epsilon_{\rm rts}}{k_{\rm rts}^{3/2}} \, . \tag{76}$$

Germano-like model According to a Germano-like approach, as an analog of (58), let us assume the STS energy dissipation can be approximated as

$$\epsilon_{\rm sts} \cong C_\epsilon \frac{k_{\rm sts}^{3/2}}{\widehat{\Delta}} \, . \tag{77}$$

This way, by exploiting the identity (68), after some calculus, the following determination for the SGS energy dissipation coefficient is obtained

$$\frac{C_\epsilon(\boldsymbol{x},t)}{\Delta} = \frac{\alpha\epsilon_{\rm rts}}{(k_{\rm rts} + \widehat{k_{\rm sgs}})^{3/2} - \alpha\widehat{k_{\rm sgs}^{3/2}}} \, . \tag{78}$$

In principle, the above dynamic procedures for determining the model coefficients C_ν and C_ϵ can be adopted independently, leading to four different model combinations. Here, only two different localized dynamic kinetic energy-based models (LDKM) are actually considered for the numerical experiments. The former one (for discussion: LDKM-B) exploits both the Bardina-like dynamic determinations (70) and (76), while the other one (for discussion: LDKM-G) uses both the Germano-like dynamic coefficients (74) and (78).

5.4 Dynamic structure model

In this section, a dynamic one-equation non-eddy viscosity SGS model is developed for the SCALES methodology. The model, recently introduced for LES (*e.g.*, (Pomraning & Rutland 2002, Chumakov & Rutland 2005)), is based on the “dynamic structure” assumption. Namely, it borrows the structure of the unknown SGS stress tensor directly from the resolved Leonard stress (60), without involving the resolved rate-of-strain tensor. The significant similarity between the SGS and the Leonard stresses, which has been observed in real as well as numerical experiments (*e.g.*, (Liu *et al.* 1994)), is exploited in the model. Thanks to this similarity, one can consider $\tau_{ij}/\tau_{ll} \cong L_{ij}/L_{hh}$, so that the following approximation holds

$$\tau_{ij} \cong \frac{k_{\rm sgs}}{k_{\rm rts}} L_{ij} \, , \tag{79}$$

acronym	*SGS stress model*	*SGS kinetic energy dissipation model*
LKM	eddy-viscosity, fixed coefficient	fixed coefficient
LDKM-G	eddy-viscosity, dynamic Germano	dynamic Germano
LDKM-B	eddy-viscosity, dynamic Bardina	dynamic Bardina
DSM	dynamic structure model	fixed coefficient

Table 1. Summary of the different SGS one-equation models.

that corresponds, in particular, to the algebraic form of the model proposed in (Pomraning & Rutland 2002).

Clearly, the solution of an additional transport equation for $k_{\rm sgs}$ is still an integral part of the modeling procedure. However, due to the non-eddy viscosity nature of the dynamic structure model, the original version (51) is used in this case. According to (79) the modeled SGS dissipation becomes proportional to $k_{\rm sgs}$

$$\Pi \cong -\frac{k_{\rm sgs}}{k_{\rm rts}} L^*_{ij} \overline{S}^{>\epsilon}_{ij} \ , \tag{80}$$

and again can show both signs. Note that the present approach does not involve the definition of any model coefficient while, for the SGS kinetic energy dissipation $\epsilon_{\rm sgs}$, the model (58) can be used together with one of the dynamic procedures discussed in Section 5.3.2.

Like for the above eddy-viscosity models, the dynamic structure model (79), coupled with the solution of the energy equation (51), provides a positive feedback mechanism that automatically stabilizes the numerical solution. However, according to some authors the dynamic structure model (for subsequent discussion, DSM), which is in some ways similar to the classical scale-similarity Bardina model (Bardina *et al.* 1983), does not provide sufficient SGS dissipation for LES and therefore should be used as part of a mixed model (*e.g.*, (Kim & Menon 1999)). Nevertheless, as already pointed out in (Pomraning & Rutland 2002), the pure DSM can be successfully used for decaying isotropic turbulence simulation. That is confirmed by the results of the present study (see next Section 7).

Finally, let us address some general issues about the use of the auxiliary variable $k_{\rm sgs}$ in SCALES, apart from the particular model implemented. When numerically solving the evolution equation for the SGS kinetic energy by means of the AWCM numerical method, owing to the non-linearity of the definition (50), additional small scales are created with respect to the solution of the primary variables. Given the adopted coupled wavelet thresholding strategy, discussed in Section 3, that automatically leads to an increase in the local grid fineness, with the unavoidable deterioration of the SCALES grid compression. As practically experienced by the authors, one can obtain a field compression comparable to that one of the CVS solution (*e.g.*, see (Goldstein *et al.* 2005)), so invalidating the use of the SGS model. To bypass the problem, as already successfully tested for local modeling based on the Lagrangian approach (Vasilyev *et al.* 2006), an artificial diffusion term can be added to the right-hand-side of the energy equation, namely $\frac{\partial}{\partial x_j}\left(\mathcal{D}_k \frac{\partial k_{\mathrm{sgs}}}{\partial x_j}\right)$. To stem the creation of small scales in k_{sgs} field, the artificial diffusion time-scale, $\Delta^2/\mathcal{D}_k$, must be smaller than the convection time-scale associated with the local strain, $\left|\overline{S}^{>\epsilon}\right|^{-1}$ that results in fixing $\mathcal{D}_k = \mathcal{C}_k\Delta^2\left|\overline{S}^{>\epsilon}\right|$, $\mathcal{C}_k$ being a dimensionless parameter of order unity. In practice, for the present numerical experiments, the value $\mathcal{C}_k = 0.1$ has been verified to suffice for the purpose.

Another important aspect that needs to be mentioned is the sensitivity of the SGS energy-based models to the initial value of k_{sgs}. Setting the initial k_{sgs} too high can result in excessive SGS dissipation leading to an incorrect energy evolution. That is particularly dangerous for transient flows, like homogeneous decaying turbulence, while it is negligible for statistically steady turbulent flows like forced turbulence, as demonstrated by the results discussed in Section 7.6. A way to make the solution less sensitive to the initial condition is under study and will be the subject of a future work.

6 Algorithm Development

In this section we briefly describe the efforts that were undertaken in the area of parallel algorithm development. In order to make a substantial progress on this front we have attacked the problems on three main fronts, namely developing efficient and easily parallelizable data structure, developing parallel version of the solver and exploring more efficient parallel implementation of adaptive wavelet transform, which are essential in improving the efficiency and scalability of the parallel solver.

6.1 Data Structures

Efficiency of adaptive solvers strongly depends on the access time to the large datasets and the ability to make efficient parallel implementation of the data structure. Three different data structures has been developed for the large datasets storage: working array, Tree and Lines. The working array is mainly used for algorithm development, due to ease of testing new algorithms. Once the algorithm is tested, we rewrite it for both Tree and Lines data structures. Thus, Tree and Lines data structures or their hybrid are primarily planned to be used for large scale simulations. Both structures have been implemented in C/C++ and Fortran 90. Tree data structure provides convenient access to a node with given coordinates and therefore it is relatively easy to merge Tree with wavelet transform, with its intrinsically random data access pattern, without significant changes of the solver code. Lines data structure, on the other hand, provides fast access to an array of nodes on a line with given $D-1$ coordinates, where D is the dimensionality of the problem. Lines data structure minimizes the amount of solver-database interactions, although it requires significant modification of

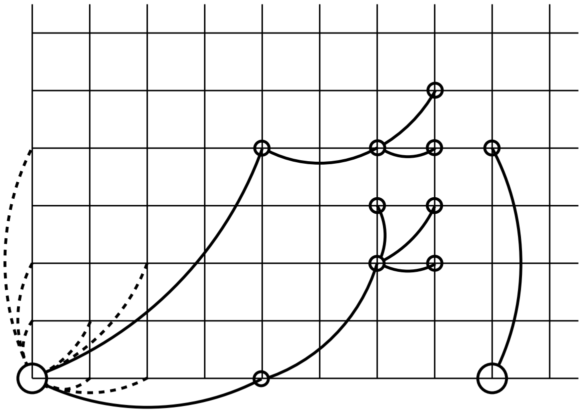


Figure 6: Tree structure for 12 nodes: root nodes (big circles) and active links (solid lines). All possible links for one of the root node are shown by dotted lines. For simplicity, we do not show the links between the nodes with the same wavelet type, level, and the type of the boundary condition.

the solver code.

6.1.1 Working Array Structure

Working array structure is used primarily to minimize the logistics of data access. Before the wavelet transform or derivative calculations, compressed data are copied into the working array, then the calculations are performed on a subset of the nodes of the working array and data are copied back to the compressed resulting array. The use of the working array data structure for large scale computations is impractical.

6.1.2 Tree Structure

The schematic diagram of currently implemented Tree structure is presented at Fig. 6. A node of level J may have up to $(J_{\max} - J)(2^D - 1)$ links to the nodes of the higher levels. Level 1 corresponds to the coarse mesh and level $J_{\max}$ corresponds to the finest mesh. Tree roots residue at the level J_{tree} which may be higher than the coarse mesh level. By setting J_{tree} to be equal to $J_{\max}$ we get a nonadaptive array storage for the mesh at the finest level of resolution. In addition to the tree structure, all the nodes belonging to the same wavelet type $(0, \ldots, 2^D - 1)$, level $(1, \ldots, J_{\max})$, and the node-type (internal, face, edge, vertex) $(1, \ldots, 3^D)$ are incorporated into a correspondent bidirectional link-list, thus forming $J_{\max}6^D$ link-lists. Single node is accessed by its coordinates. The access operation takes at most $J_{\max} - J_{\text{tree}}$ steps along the tree and requires $(J_{\max} - J_{\text{tree}})D$ comparisons for the correspondent bits of the node coordinates. A group of the nodes of the same wavelet type, level, and relation to the domain boundary is accessed through the traversing the correspondent link-list.

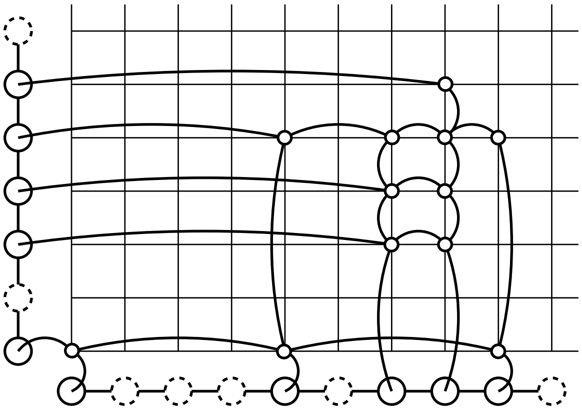


Figure 7: Lines for 12 nodes: Face nodes (big circles) and active links (solid lines). For simplicity, we show all the nodes on a line linked together; in reality, separate link-lists are created for the nodes of different levels.

6.1.3 Lines Structure

The schematic diagram of currently implemented Lines data structure is presented at Fig. 7. A face is introduced for each dimension D as an array of face nodes of the dimension $D-1$. Each face node contains the beginnings of $J_{\max}$ one-directional link-lists. Each link-list joins the nodes of the same level on the given line. Nodes are accessed in groups only. A typical access operation will copy the data associated with the nodes of a given line and given level into a corresponding array. After being processed by a specially designed version of wavelet transform, the whole line of the nodes is written back to the Lines data structure.

6.2 Algorithmic Optimization of the Parallel Wavelet Transform

When parallelizing the solver, the main cost is associated with adaptive wavelet transform, since the wavelet transform requires synchronization of wavelet coefficients at one level prior of performing next level wavelet transform. This makes load balancing extremely hard, since load balancing is required at each level of resolution. One way to achieve efficient parallelization is to modify wavelet transform in such a way that it can be performed anarthrously. This way the load balancing only depends on the total number of grid points in the domain and inter-processor communication cost. Five different algorithmic modifications of parallel wavelet transform are investigated. In order to avoid modification of the actual code, the non-adaptive wavelet transforms were considered. Once the strategy was developed, it was implemented in actual adaptive wavelet transform used in the adaptive wavelet collocation solver. The algorithmic modifications are discussed next. In order not to complicate the discussion, the algorithms are discussed in one dimension, while the actual implementation was carried out in two dimensions.

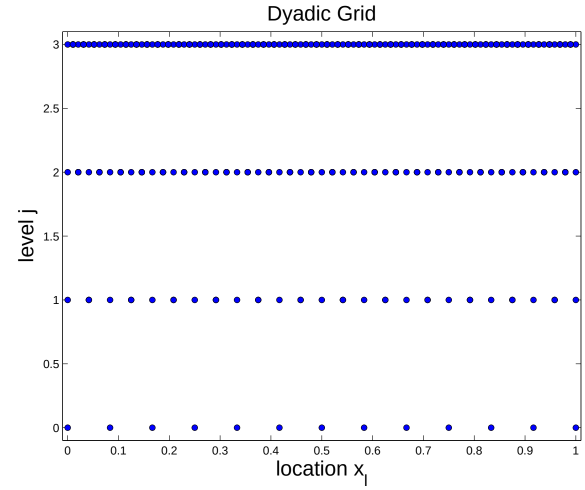


Figure 8: The dyadic grid in one dimension, which shows the multi-level assignment of the points used in the wavelet transform.

6.2.1 Serial Wavelet Transform

The wavelet transform is currently being used in the AWCM to dynamically adapt the grid over which a simulation is performed. It has proven to be a successful way to use computational resources efficiently while obtaining accurate solutions to complex fluid flow systems. The wavelet transform is performed on a dyadic grid, which is shown in Figure 8. This grid separates the domain into multiple levels of resolution. All points are initially placed on the finest level of resolution ($j = 3$ in the figure). Then, every other point (an even point) is carried down to the next lower level, leaving all the odd points at the higher level.

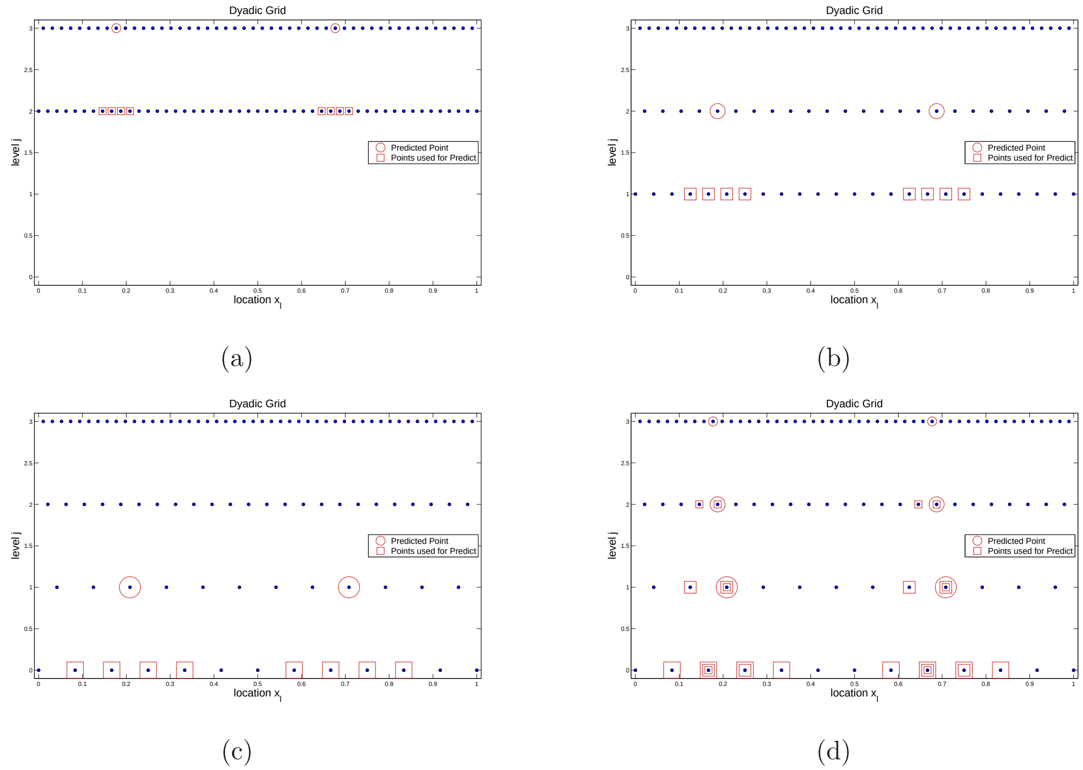
$$x_k^j = x_{2k}^{j+1} \tag{81}$$

This step is repeated until the lowest level of resolution is achieved. Therefore, every level has twice as many points as the next lower level. The resolution difference between the highest and lowest levels of resolution is vast, which accommodates all sizes of structures that may exist in the system. For visualization reasons, the dyadic grid will show points only at their lowest levels of resolution for the rest of this report. Therefore, the resolution at a level is actually twice better than what is shown.

The simplified forward wavelet transform is as follows:

$$d_k^j = \frac{1}{2}\left(c_{2k+1}^{j+1} - \sum_l w_{k,l}^j c_{2k+2l}^{j+1}\right) \tag{82}$$

$$c_k^j = c_{2k}^{j+1}. \tag{83}$$



(a)

(b)

(c)

(d)

Figure 9: The predict stage of the forward wavelet transform assuming 4th order interpolating wavelets. A circled point is predicted using the interpolated value calculated from the associated squared points on the next lower level. The lower right plot shows the first three plots all put together.

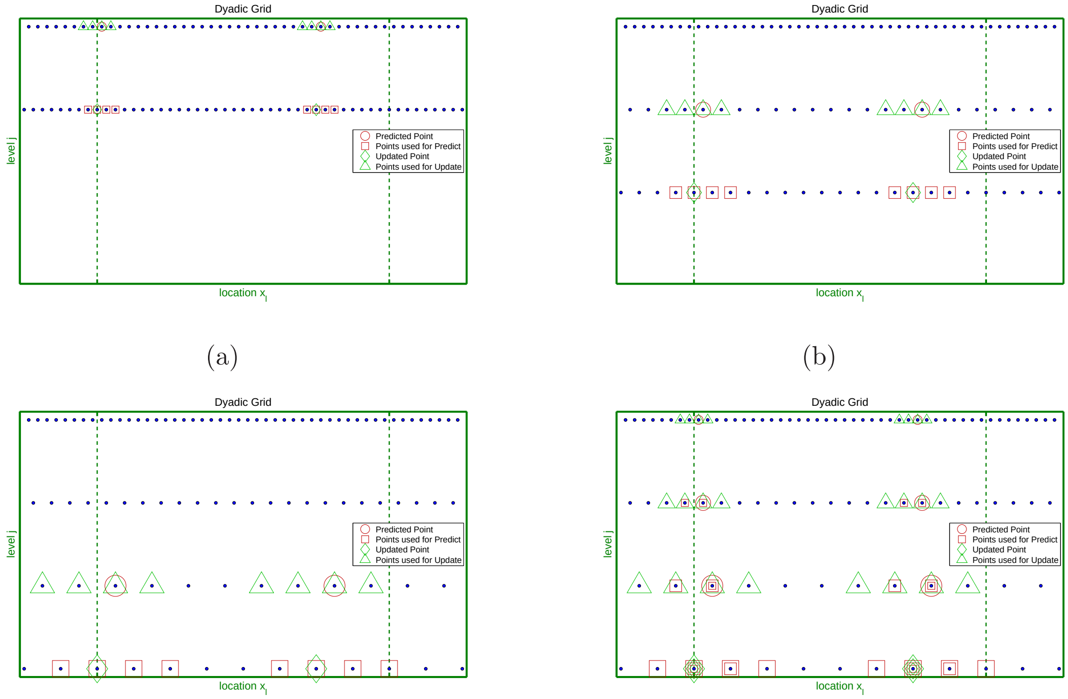
Here, the c_k^j values are the function values at location k on level j. The $w_{k,l}^j$ are weights associated with the wavelet interpolating functions. Therefore, a wavelet coefficient, d_k^j, is a measure of the difference between the function representation at a level and its approximation (using the wavelets) at the next lower level. This is called a predict stage, since the wavelet coefficient is calculated by predicting the function value using the interpolated points on the next lower level. The predict stage of the wavelet transform is illustrated in Figure 9.

The description of the forward wavelet transform given above is a simplified case. The full forward wavelet transform includes an update stage following the predict stage.

$$d_k^j \quad = \quad \frac{1}{2}\left(c_{2k+1}^{j+1} - \sum_l w_{k,l}^j c_{2k+2l}^{j+1}\right) \tag{84}$$

$$c_k^j \quad = \quad c_{2k}^{j+1} + \sum_l \tilde{w}_{k,l}^j d_{k+l}^j \tag{85}$$

The predict stage remains the same, but the c-values that get carried down to the next lower level of resolution are updated using the wavelet coefficients that were calculated during the predict stage. This process is illustrated in Figure 10. The update stage guarantees that the wavelet interpolating functions have zero mean, which ensures an accurate transform.



(a)

(b)

(c)

(d)

Figure 10: The predict (red) and update (green) stages of the forward wavelet transform. A circled point is predicted using the interpolated value calculated from the associated squared points on the next lower level. A point with a diamond is updated using the triangular wavelet coefficients on the next higher level of resolution. The lower right plot shows the first three plots all put together. The dark green vertical lines represent a hypothetical domain decomposition. The local subdomain consists of the points within the two lines and the points outside the lines make up a buffer zone added for accurate computation.

In fact, the interpolating wavelets of order N when using the update stage have N vanishing moments. This allows for faster implementation of the wavelet transform by a factor of two. Also, with the addition of the update stage, the inverse wavelet transform is simple to implement. The order of operations are reversed, and the inverse wavelet transform is performed from low to high levels of resolution as opposed to high to low for the forward wavelet transform. However, the drawback is that the inter-level connection required by this update stage is a main issue in parallelizing the wavelet transform.

For problems of arbitrary dimension (greater than one), the wavelet transform is performed by transforming each dimension independently. As the levels of resolution are descended (or ascended in the inverse wavelet transform) in one dimension, the transform is completed over the entire domain at that level of resolution. Therefore, in the three dimension case, the transform is completed first in the x-direction. During the j-th level of the x-direction transform, the points in the y- and z-directions that are transformed are also on the j-th level of resolution. Once the x-directional transform in complete, the wavelet transform is performed, in a similar fashion, in the y-direction, and finally in the z-direction.

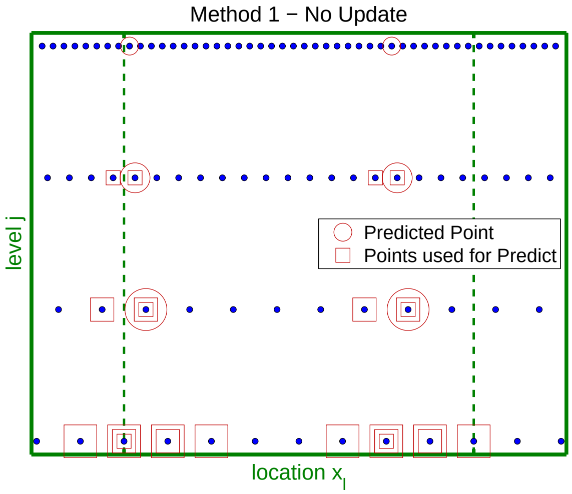


Figure 11. An illustration of the No Update Method. This method skips the update stage over the entire domain.

6.2.2 Parallel Wavelet Transform

In order to completely and accurately perform the update stage of the wavelet transform (obtaining the correct c-values at each level of resolution), the points must be synchronized across subdomain boundaries. This is shown in Figure 10 by the dark green subdomain boundaries. The points lying outside of the boundaries are buffer zones added for proper interpolation. At lower levels of resolution, the cascading of this synchronization becomes even greater. At some level (depending on processor count and global domain size), the entire domain must be synchronized on all processors to complete the update stage. The synchronization is necessary once for every level of resolution (after each update stage) and also for every dimension. Therefore, for a three dimensional problem with 5 levels of resolution, 15 communication stages are required to simply perform one forward wavelet transform. The inverse wavelet transform requires the same amount of stages resulting in 30 synchronization stages for one time step while using the AWCM. It is expected that the cost of so much communication is likely to be too expensive.

Figure 10 illustrates the amount of interplay between subdomains. At each level of resolution, the points near the boundaries (the left point in the figure) must have access to points in the neighboring processor's local subdomain. At lower levels, the points required from neighboring processors are far into the domain on the finest level of resolution. The addition of the update stage requires the use of all points on every level, up to the finest level of resolution, associated with the neighboring subdomain's points, in order to have an accurate value when the local subdomain points near the boundaries use them. Therefore, either a lot of communication is necessary, or the wavelet transform must be modified so that the interplay between subdomains has less of an effect. We have considered five different parallel implementations with the idea that the performance of these different methods will

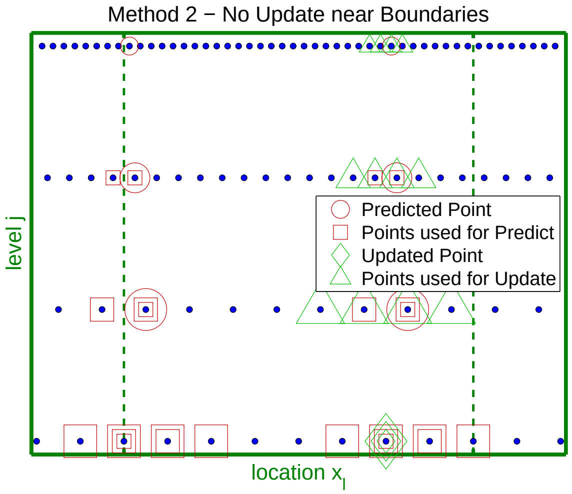


Figure 12: An illustration of the No Boundary Update Method. This method skips the update stage only for points near the local subdomain boundary. If a local point needs a point from a neighboring subdomain, the update stage is not performed.

shed light on how to modify the wavelet transform so that its parallel implementation is fully optimized.

6.2.3 Five Algorithmic Modifications

The five algorithmic modification of the parallel wavelet transform that are studied all have their own unique advantages and disadvantages. Some of the characteristics are fully anticipated, while others have yet to be investigated.

The first method should be considered as a base case, and it involves not performing an update stage over the entire global domain. This implementation will be referred to as the No Update Method. This method is currently implemented in actual adaptive wavelet transform reported in Section 6.3. The buffer zone exists on each subdomain so that the predict stage can be properly implemented. This requires a small amount of extra computation for problems in multiple dimensions, since the buffer zone must also be transformed in all but the final dimension so that the proper values are available when the local subdomain needs them during the predict stage. However, due to the lack of the update stage, the computational time is also cut down since the algorithm takes half as many steps. Also, there is no inter-level communication required. The results from the No Update Method should exactly match the results from the serial case with no update. The main drawback for this method is a loss of zero-mean properties of the interpolating wavelet. The No Update Method is illustrated in Figure 11.

The second implementation reduces the disadvantages of the No Update Method. The second method performs the update stage within the local subdomain. For points that would require wavelet coefficients from outside the local subdomain during the update stage, no

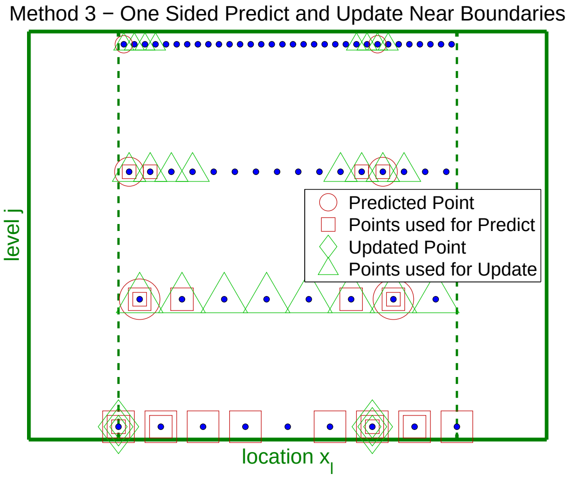

Method 3 − One Sided Predict and Update Near Boundaries
level j
location x_i
○ Predicted Point
□ Points used for Predict
◇ Updated Point
△ Points used for Update

Figure 13: An illustration of the One Sided Transform Method. This method segregates all processors from each other. All subdomains are treated as a small global domain with one sided predict and update stages near the boundaries.

update is performed. Therefore, this method is called the No Boundary Update Method. The second is more difficult to implement in adaptive wavelet transform due to increased level of bookkeeping. However, the No Boundary Update Method method is a candidate to be used in actual adaptive wavelet transfrom, once No Update Method is implemented and fully optimized. It is an improvement from the previous method since the zero mean properties are kept for the wavelets in almost the entire domain. However, near the boundaries, the update stage is not performed and the wavelets have no vanishing moments. As No Update Method the No Boundary Update Method requires no inter-level communication, and there is a slight drop in computational cost due to the increased buffer zone near the boundary that need to be synchronized only at the beginning of the transform. Figure 12 shows the No Boundary Update Method.

The third method improves the excess computational costs from the previous methods by completely segregating each local subdomain. Therefore, each processor's subdomain is treated as in the serial case with a small global domain. There is no connection to other regions of the actual global field. If a point does not exist in the local subdomain, it has no impact on the wavelet transform for that processor. One sided predict and update stages are performed near the subdomain boundaries, just as would occur near the boundaries of the global domain in the serial case. Therefore, this implementation will be referred to as the One Sided Transform Method. The huge advantage for this method is the low computational cost due to the lack of a buffer zone. There is also no inter-level communication, and the update stage is performed over the entire domain (although one sided near the boundaries) ensuring the zero mean properties for the wavelets. The main drawback of this method is that it results in discontinuous derivative at the subdomain boundaries. As a result the method is not a candidate for use in actual adaptive wavelet collocation solver and the inclusion of

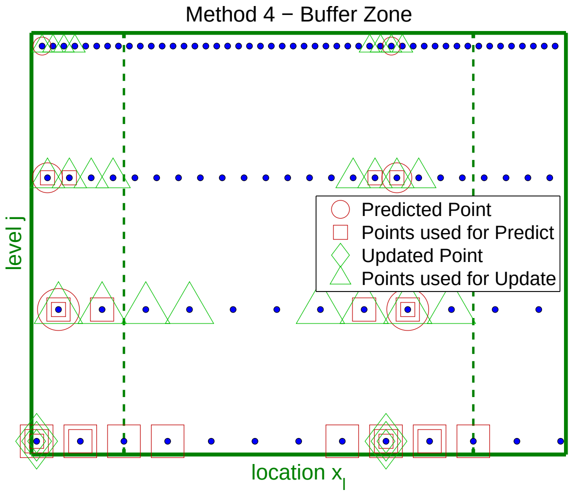


Figure 14: An illustration of the Buffer Zone Method. This method uses a buffer zone that contains points from neighboring subdomains. The buffer zone points are transformed using a one sided predict and update.

it into this study is of pure academic interest. Figure 13 gives a representation of the One Sided Transform Method in one dimension.

The fourth method extends the idea of the One Side Transform Method to include a buffer zone. For this method, referred to as the Buffer Zone Method, the information from neighboring subdomains is available to the points in a local subdomain. This should help with accuracy and the negative effects of the one sided transform near the boundaries of the local subdomain and could be used in connection of Schwartz domain decomposition iterative method. Extra computation is required since the update stage is performed everywhere, including the buffer zone where a one sided transform is carried out. Therefore, the work required to perform the update stage within the buffer zone is also being done in the neighboring subdomain. This method improves the loss of derivative continuity properties, however as in the previous method does not guarantee the continuous derivatives across the subdomain boundaries. In the Buffer Zone Method, the update stage is performed over the entire domain, keeping the zero mean wavelet properties, and no inter-level communication is required. The Buffer Zone Method is displayed in Figure 14.

The final implementation may be considered as another base case. For this method, the appropriate buffer zone points are passed to neighboring processors after each update stage in every dimension. The results obtained from this Complete Communication Method should exactly match those obtained from the serial case, even with an update stage. However, it is anticipated that the communication cost from the inter-level inter-dimension synchronization of the local subdomains will be severe. The Complete Communication Method is demonstrated in Figure 15. Due to its high computational cost this method is not a candidate for implementation in parallel adaptive wavelet collocation code.

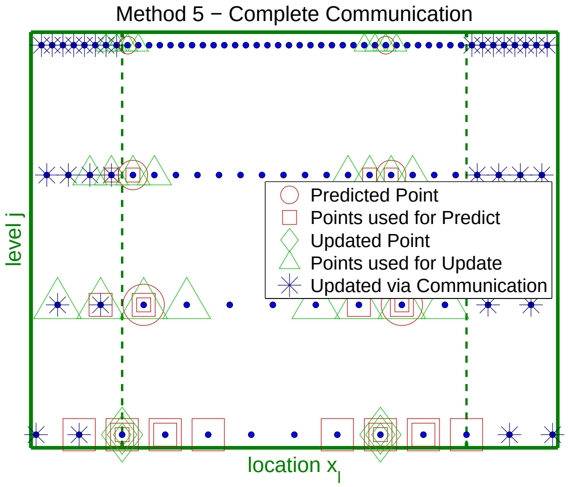


Figure 15: An illustration of the Complete Communication Method. Communication of the buffer zone is completed after each update stage in every dimension. This method will obtain the exact same solution as the serial case.

6.2.4 General Problem Description

When studying the performance of different implementations of parallel wavelet transforms, a few restrictions were placed on the global system. First, for simplicity, the number of processors must evenly divide the d-dimensional global field in a checkerboard type decomposition. Therefore,

$$p^{\frac{1}{d}} = \text{integer}. \tag{86}$$

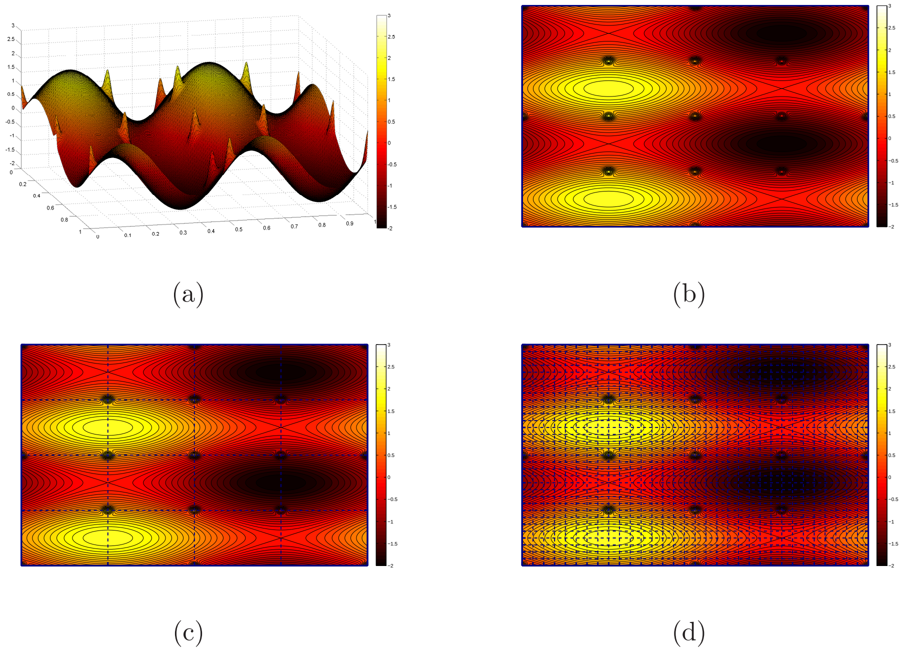
Second, as another simplifying assumption, the number of processors in one dimension should evenly divide the number of points on the coarsest level in one dimension.

$$\frac{M_i - 1}{p^{\frac{1}{d}}} = \text{integer}. \tag{87}$$

This ensures that each processor has an even number of coarsest-level intervals. Lastly, the problem is also made simpler by restricting the minimum number of coarsest-level intervals so that the nearest neighboring processors contain the points needed for both the predict and update stages.

$$\frac{M_i - 1}{p^{\frac{1}{d}}} \geq \max\left[n_{\text{predict}} - 1, n_{\text{update}}\right], \tag{88}$$

where the n values are the number of points on each side used for the interpolation during the predict and update stages. The order of the wavelets is then $2n$. Therefore, a processor does not need to communicate with more than one other processor for each boundary of its local subdomain. With all of these restrictions in place, for a two dimensional field, as used for all cases reported here, it was found that the processor count, p, could only take only



(a) (b) (c) (d)

Figure 16: The On-Boundary Function. A surface plot in the upper left shows the sharp localized peaks overlaid on the smooth large structured sinusoids. The other three images show a contour plot for processor counts of 1, 16, and 1024. The processor boundaries cut through every one of the sharp peaks.

even powers of 2. This is a result of restricting the total number of points in each direction (N) and the number of coarsest level points in each direction (M) to powers of 2. So,

$$M_i - 1 = M - 1 = 2^m \tag{89}$$

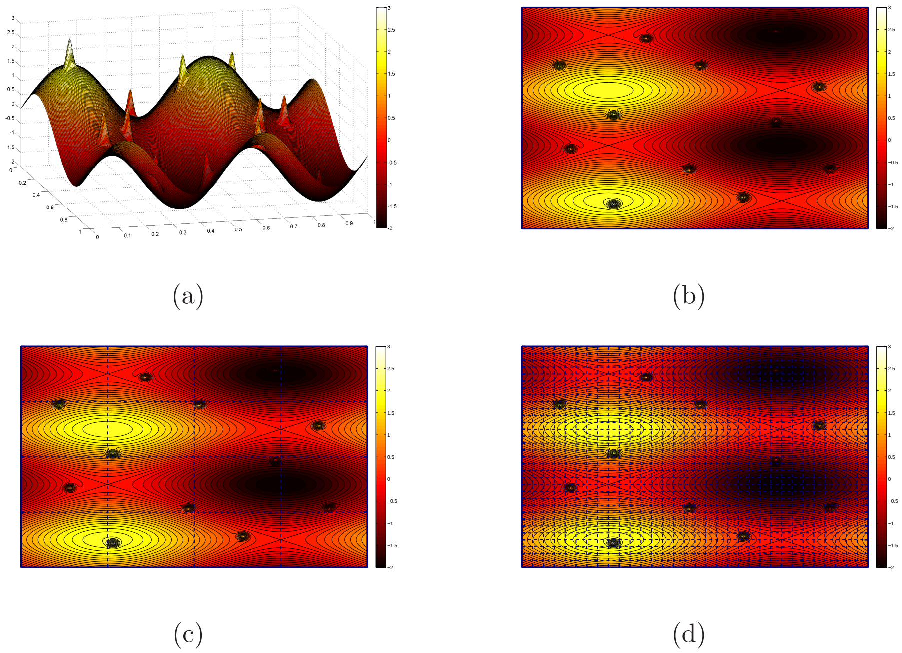
$$N_i - 1 = N - 1 = 2^n \tag{90}$$

From this restriction, setting the finest level of resolution is trivial:

$$j_{\max} = n + 1 - m, \tag{91}$$

where the coarsest level of resolution is $j = 1$. It is easy to see that when $n = m$, there is only one level of resolution and no transform will be performed.

The fields for which the wavelet transform acts on is an important consideration. The field should have some regions with sharp transitions and other areas where the structures are not localized. The placement of the sharp transitions may affect the ability of certain methods to accurately perform its modified wavelet transform. Therefore, two functions are tested. Both fields have sinusoids in the x- and y-directions with sharp Gaussian peaks dispersed throughout the domain. The sinusoids act as large structures, while the sharp Gaussian peaks represent localized structures that are well compressed by the wavelet transform.



(a) (b)

(c) (d)

Figure 17: The Off-Boundary Function. A surface plot in the upper left shows the sharp localized peaks overlaid on the smooth large structured sinusoids. The other three images show a contour plot for processor counts of 1, 16, and 1024. The peaks are offset from all the processor boundaries so that none of the peaks are split between local subdomains.

For both functions, the total number of Gaussian peaks overlaying the sinusoids is twelve. The difference between the two functions is the placement of the Gaussian peaks. In the first function, the peaks are located directly on the processor subdomain boundaries. This function is referred to as the On-Boundary Function (func=2 in the plots in Section 6.2.5) and is illustrated in Figure 16. The second function sees a shift in the peaks so that no processor boundary goes through any of the peaks. This function is then referred to as the Off-Boundary Function (func=4 in the plots in Section 6.2.5) and is displayed in Figure 17. In the Off-Boundary Function, no peak is shared amongst multiple subdomains, even for large processor counts.

The general process for this study is to take our field and perform the appropriate modified forward wavelet transform for each of our five implementations. Then, the wavelet coefficients are analyzed and those that are deemed insignificant via the ϵ thresholding ($d_k^j < \epsilon$) are set to zero ($d_k^j = 0$). Next, the appropriate inverse wavelet transform in applied to get the functional values once again. The accuracy can then be tested by calculating the error between the initial field and the resulting field after the inverse wavelet transform. Both L_∞ and L_2 errors are reported. Another related and important characteristic for the transform is the compression. Since the wavelet transform is used for data compression and adapting grids, a high level of compression is always attractive. The compression will be calculated

5 Methods	6 Processor Counts	2 Function Types	3 Fixed Global Field Sizes	3 Fixed Local Field Sizes
1.) No Update	1	On-Boundary	257×257	9×9
2.) No Boundary Update	4	Off-Boundary	513×513	33×33
3.) One Sided Transform	16		1025×1025	129×129
4.) Buffer Zone	64			
5.) Complete Communication	256			
	1024			

Table 2: This table lists all the runs performed for the parallel wavelet transform. A total of 340 runs were completed in parallel mode.

as follows:

$$1 - \text{Compression} = \frac{N_{\geq}}{N_{\text{total}}}, \tag{92}$$

which is the ratio of significant wavelet coefficients to the total number of points. The timing performance of the implementations is also investigated. Therefore, the process of forward wavelet transform, ϵ thresholding, and inverse wavelet transform is performed for a number of iterations to ensure accurate timing statistics. The average time per iteration is calculated and reported. Also, since communications are involved, the first iteration is not timed so that all communication links have an iteration to get warmed up.

All reported runs used a two dimensional field with $\epsilon = 10^{-6}$. The wavelets used were 4^{th} order by setting $n_{\text{predict}} = 2$ and $n_{\text{update}} = 2$ ($n_{\text{update}} = 0$ when no update is performed). Also, the m value was chosen so that the finest level of resolution was always 3. Therefore, two transform steps are taken in each dimension for each wavelet transform. For the reported results the main parameter is the method type and the processor count. The function type and the field sizes were also varied. Table 2 gives a complete summary of all the parallel cases that were run. Table 3 then lists the types of serial cases that were completed. All runs were completed on the BG/L machine.

6.2.5 Scalability, Compression, and Accuracy

Many runs for each of the five implementations were completed and compiled with the main goal of comparing the different methods and their characteristics. Of importance are the timing performance, accuracy, and compression as processor count is increased, type of function is varied, and field size is increased. One important note is that in actual adaptive wavelet transform the timing performance would greatly depend on the compression. Since this study focuses on the non-adaptive wavelet transform only, the compression value is only a statistical property that does not affect the computational cost. A point that is deemed insignificant via the ϵ thresholding still has its functional value found by the inverse wavelet

2 Methods	1 Processor Count	2 Function Types	3 Fixed Global Field Sizes	3 Fixed Local Field Sizes
A.) Without Update	1	On-Boundary	257 × 257	9 × 9
B.) With Update		Off-Boundary	513 × 513	33 × 33
			1025 × 1025	129 × 129

Table 3: This table lists all the runs performed for the serial wavelet transform. A total of 24 runs were completed in serial mode.

transform.

In looking at all the plots, it is important to remember that all implementations should be compared against the appropriate serial case. Therefore, the No Update Method should be compared against the serial case with no update, whereas the rest of the methods should be compared against the serial case with the update stage. Also, when comparing the results for the different types of functions, the change in the difference from the serial case is more important that the absolute difference between the two function cases.

In terms of timing performance, the results match expectations fairly closely. Figure 18 shows the average per iteration timing for a small fixed local field size for the Off-Boundary Function. Overall, it shows an obvious disparity between most of the methods. The green line corresponds to the One Sided Transform Method. This method remains very close to the average time per iteration for a single processor. Since the local field size is fixed for this plot, and the One Sided Transform Method has no buffer zone to add extra computation, the time does not increase. The red line is the No Update Method. This method is a little slower than the One Sided Transform Method, since it has a buffer zone that adds extra computation. However, it does not do an update stage, so some of the computational cost is given back. The blue and purple lines overlap in this figure. These two lines correspond to the No Boundary Update Method and the Buffer Zone Method. Both these methods have a buffer zone, which adds computational cost. The Buffer Zone Method performs an update stage in the buffer zone, whereas the No Boundary Update Method does not. Therefore, upon zooming in on the plot, the Buffer Zone Method is slightly slower. Finally, the cyan line, representing the Complete Communication Method, shows a much worse performance for this implementation. This method performs an update stage everywhere and, therefore, has the same computational cost as previous methods. The extra time can be attributed to the high level of communication necessary to synchronize the buffer zone after each update stage in each dimension.

The speed up and efficiency are also studied. Figure 19 illustrates those characteristics for each method for all three fixed global field sizes. As expected, the performance shown in these plots matches the order in the average per iteration timing analysis. At large processor counts, the speed up appears to be growing linearly for all methods, just with

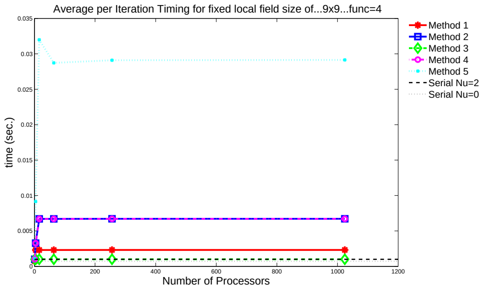


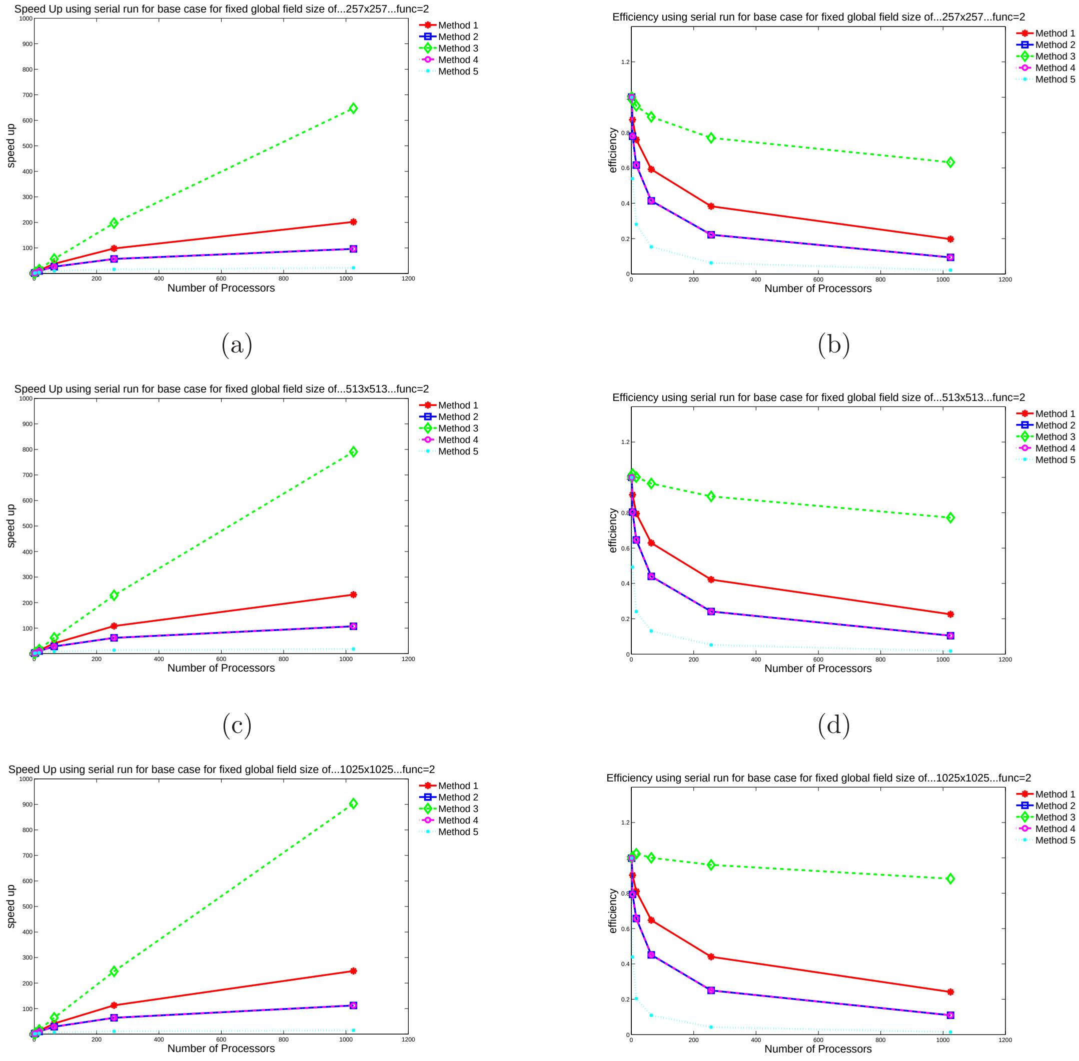
Figure 18: Plot of average per Iteration Timing for a small local fixed field size. All the timing plots look very similar to this, with an obvious separation between the methods.

different slopes. As the fixed global field size is increased, the spread between the methods grows. Therefore, for larger domains, the One Sided Transform Method does better than the rest at a growing rate. The Complete Communication Method is the only method that has a decreasing efficiency as the global field size increases. This is due to the fact that this method must communicate a great deal of data many times for each wavelet transform. For larger global field sizes, the amount of data that must be sent is increased, which in turn increases the time required to complete an iteration. Thus, efficiency is lowered.

The type of function had essentially no effect of the timing performance for all methods. This can be attributed to the fact that the transform is being performed non-adaptively. The same number of computations and communications will take place regardless of the function used and the value of wavelet coefficients after the forward wavelet transform.

Unlike the timing performance, the accuracy and compression could change with a dependence of the type of function. Figure 20 uses the L_2 norm error to represent the accuracy for a large fixed global field size. As stated previously, the L_2 norm is a measure of the average error over the entire domain. The figure displays the accuracy for both the On-Boundary Function (func=2, left) and the Off-Boundary Function (func=4, right). The No Update Method (red) has the worst error since there is no update in the entire domain. The lack of a vanishing moment is causing a loss in accuracy. This method does follow closely to the no update serial case, as expected.

The next best method is the No Boundary Update Method (blue). As the processor count increases, the error gets worse. The lack of an update stage near the boundaries is causing a loss in accuracy. The Buffer Zone Method (purple) follows very closely to the Complete Communication Method (cyan), which gives exactly the same results as the serial case (with update stage). The added buffer zone is doing a good job maintaining the accuracy. This also acts as a check that the Complete Communication Method is working properly. The One Sided Transform Method (green) actually does the best at very high processors. It has the lowest error with 1024 processors and competitive errors for lower processor counts. In comparing the two function types, almost all the methods keep their same difference from



(a)

(b)

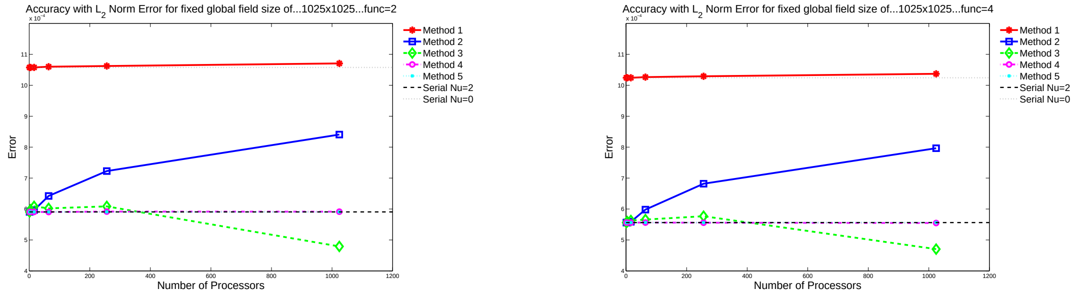
(c)

(d)

(e)

(f)

Figure 19: Speed Up and Efficiency plots for increasing fixed global size. The scales remain the same for all three field sizes. There is an overall spreading between the best and worst method performance as field size increases.

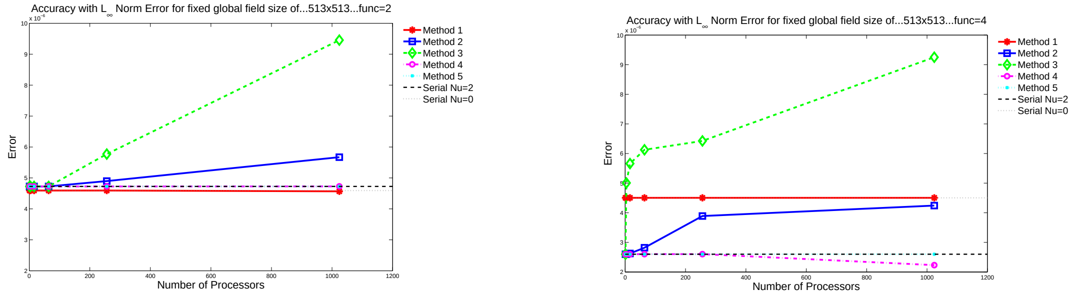


(a) (b)

Figure 20: The accuracy using the L_2 norm error is compared for the implementations and for the two functions. In comparing dependence of accuracy of the type of function, one must view how far a method is from the serial case. Method 1 should be compared with the serial case with no update. All other methods should be compared with the serial case with the update stage.

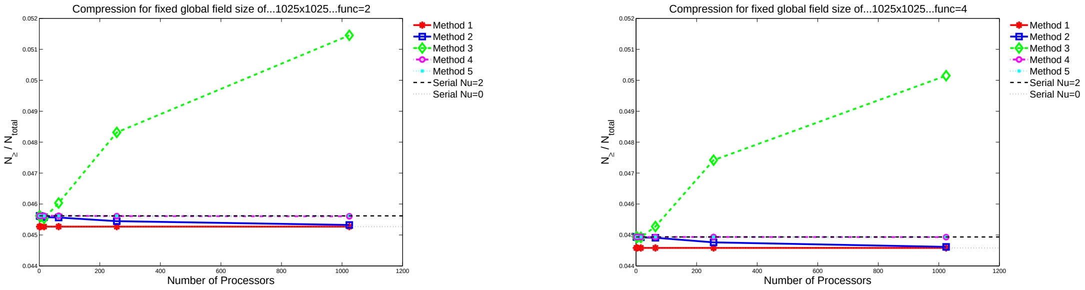
the appropriate serial case. This signifies no dependence on type of function, since that particular method has a similar accuracy as compared to the serial case for both functions. As compared with the serial case (with update stage), the green line shows a worse accuracy for the Off-Boundary Function.

Figure 21 is another accuracy plot, except this one uses the L_∞ norm error, which is a maximum error for the domain. All methods perform similar to the L_2 norm error accuracy plot except for the One-Sided Transform Method. Whereas in Figure 20 the One Sided Transform Method appears to have one of the lowest errors, here the green line towers above the rest. This signifies high local errors, despite having a low average error. The high local errors most likely occur near the local subdomain boundaries where a one sided transform is being used. It appears that doing one sided transforms well within the global boundaries and having no interconnection between subdomains is causing high local errors. Again, it does worse for the Off-Boundary Function. It may have more trouble accurately interpolating



(a) (b)

Figure 21: The accuracy using the L_∞ norm error is compared for the implementations and for the two functions. In comparing the dependence of the accuracy on the type of function, one must view how far a method is from the serial case. Method 1 should be compared with the serial case with no update. All other methods should be compared with the serial case with the update stage.



(a) (b)

Figure 22: The compression is compared for the implementations and for the two functions. In comparing the dependence of the compression on the type of function, one must view how far a method is from the serial case. Method 1 should be compared with the serial case with no update. All other methods should be compared with the serial case with the update stage.

the tails of the sharp Gaussian peaks (Off-Boundary Function) with the one sided transform at the local subdomain boundary than the center of the peaks, which occurs for the On-Boundary Function. The Buffer Zone Method in the Off-Boundary Function plot actually performs better than the Complete Communication Method and therefore the serial case.

The final wavelet transform characteristic to be investigated is compression. The compression statistics for a large fixed global field size and for both types of function are shown in Figure 22. All of the methods show similar compression values except for the One Sided Transform Method, which has a relatively large ratio of significant points to total points. This is an unattractive characteristic, as the lower this ratio is, the greater memory saved once the adaptive wavelet transform is implemented. The lower compression could account for the average error being less than the other methods. The One Sided Transform Method has more points using their actual wavelet coefficient values, lowering the average error. However the high local errors from the L_∞ norm accuracy plot cannot be justified. A low compression can only account for lower accuracies. This figure also shows that all methods have a common shift in comparing the two types of functions (even the One Sided Transform Method). The shifts match the shifts in the serial cases. It appears that the difference between these two types of functions does not affect the compression.

6.3 Parallelization of DAWC Method

When parallelizing the code the following important issues need to be considered: asynchronous events on different processors, concurrency and possible deadlocks. In order to facilitate faster parallel code development the following strategy has been used. The code was designed to consist of three main parts: the main wavelet code, the data structure, and the parallel communicators with the required interfaces and external libraries (Fig. 23).

Keeping these three parts separate simplifies code structure and facilitates debugging, though inhibit code performance. In addition, all the information transfers between pro-

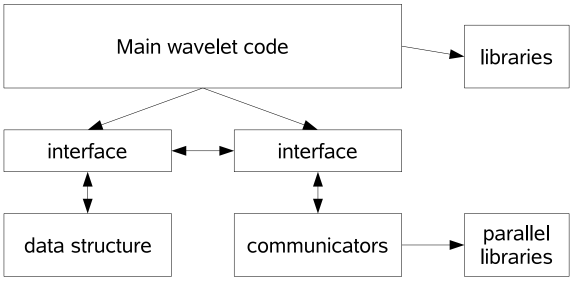


Figure 23. Code structure. Arrows represent the direction of function calls.

cessors were implemented via collective all-to-all MPI (message passing interface) communications, which eliminated all the deadlock related problems. The major trade-off of such an approach was, again, the code performance. Normally, a processor would require information from several neighboring domains only, not from all the other processors, therefore replacing all-to-all with point-to-point communications is an important optimization which will be implemented in the future.

6.3.1 Input/Output Ideology

Our parallel adaptive wavelet based code as well as visualization tool can be run on several processors using MPI library. Input/output ideology is the following: if we are using N processors we expect to read single input file and, if necessary, N result files. The code running on N processors will produce N output files (.res, .log, .vtk, or others). An extension is added to output files to indicate the processor number that file has been written by (e.g. p000, p001, p002, etc).

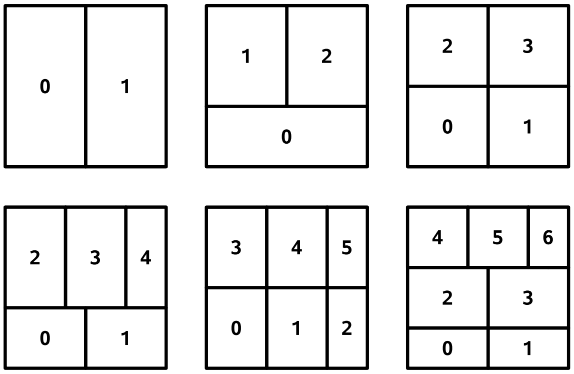


Figure 24: Geometric decomposition of 8×8 domain of trees for different number of processors N. First subdivision occurs perpendicular to the vertical axis on $\sqrt{N}$ sub-domains.

6.3.2 Domain Decomposition

We have implemented a geometric domain decomposition based on the maximum possible number of wavelets inside each of the sub-domains. In short, a rectangular d-dimensional computational domain has to be distributed among a given number, N, of processors for a given set of active wavelets. First of all, the domain is subdivided perpendicular to the first axis on rounded to the nearest integer $\sqrt[d]{N}$ sub-domains. The available N processors are distributed among these sub-domain according to the number of active wavelets inside each of the sub-domains. Then, the first step is repeated, i.e. each of the previously obtained sub-domains is subdivided perpendicular to the second axis on rounded to the nearest integer $\sqrt[d-1]{N'}$ number of sub-sub-domains, where N' is the number of processors previously assigned to that particular sub-domain. This recursion steps are repeated d times and results in a quite balanced decomposition of the initial domain. An example of decomposition of 2-D square domain is presented at Fig. 24 schematically. The actual mesh and corresponding solution are shown in Figs. 25-27.

User can control domain decomposition via the input parameter file. Domain splitting may happen only along the predefined directions. For example, providing three dimensional parameter `domain_split = 0,1,0`, where non-zero component allows domain subdivision in that direction, would force domain decomposition into slabs perpendicular to y-axis.

6.3.3 Load Balancing

Load balancing is closely connected to the domain decomposition. Currently, we do not perform any load balancing. Initial domain decomposition is performed based on the regular grid of active wavelets and is not updated while the code runs.

This will not be the case in the nearest future. Load balancing is an important issue to address. The computational load of a processor nearly linearly depends on the number of active wavelets inside that processor domain. Poorly loaded processor inevitably creates a significant source of bottlenecking and drastically impedes code performance.

We plan to start by implementing a simple approach that would perform domain decomposition after a given number of time-steps. Then trees, which are the finest elements the decomposition deals with (we will consider the trees in details in Section 6.3.4), are to be transported to their newly assigned processors. Another approach would be to perform new domain decomposition and trees reassignment in alternating direction one at a time, i.e. in x-direction, then after several time-steps in y-direction, etc. This could minimize communications between the processors inside load balancing subroutines.

More complicated procedures have to be implemented for the distributed systems with non-uniform communication delay or memory access. Neighboring processors are to be assigned taking into account communication delays as well as the number of active wavelets inside each processor's domain has to be related to the performance of the current processor.

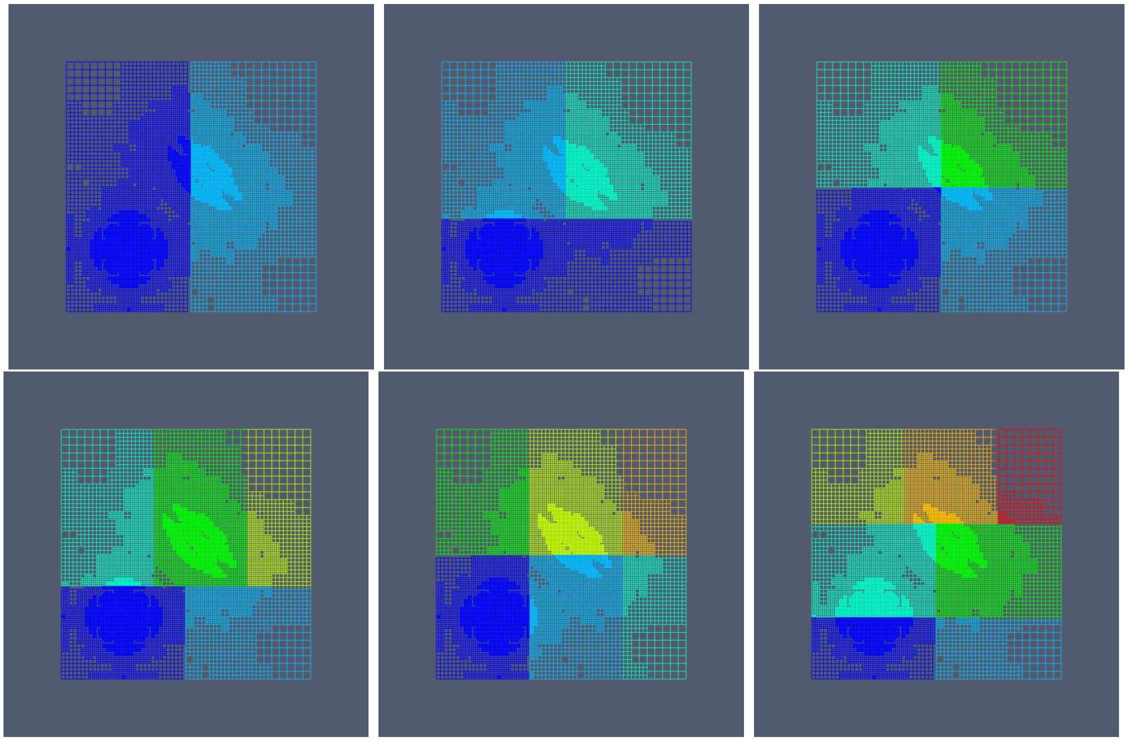


Figure 25: Geometric decomposition of 8 × 8 domain of trees for different number of processors N. First subdivision occurs perpendicular to the vertical axis on $\sqrt{N}$ sub-domains. Grid is shown.

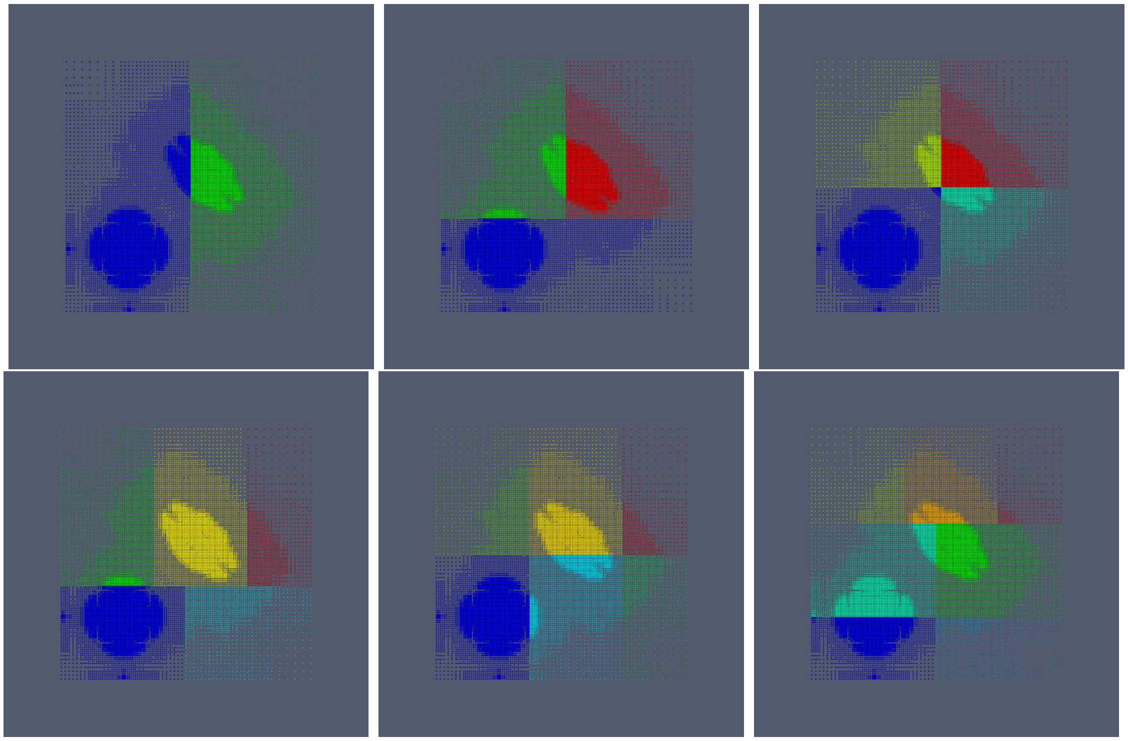


Figure 26: Geometric decomposition of 8 × 8 domain of trees for different number of processors N. First subdivision occurs perpendicular to the vertical axis on $\sqrt{N}$ sub-domains. Wavelets are shown.

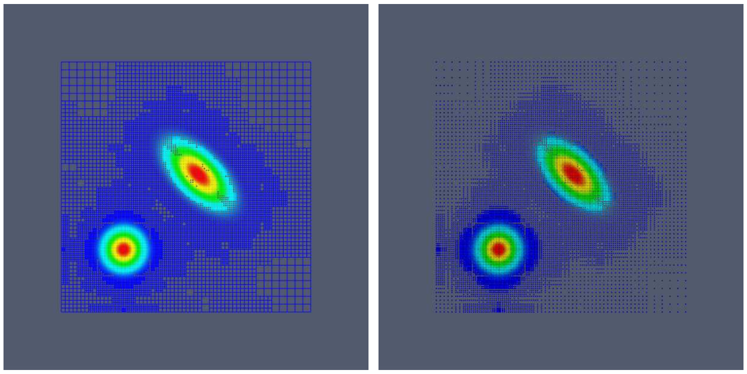


Figure 27. Solution: grid and wavelets.

Efficient load balancing is currently under investigation.

6.3.4 Parallel Code Structure

The code consists of three main parts: the data structure, the parallel communicators, and the main wavelet code (Fig. 23).

In the current implementation each processor's data structure contains a regular array of quad-tree roots of the whole domain initiated at level j_{root}. When a new node of some level j to be added to the domain, a link is inserted into the correspondent quad-tree node of level $j-1$. Maximum access time of any node through such a structure is obviously linear to the maximum level of wavelets, i.e. $O(j_{\max} - j_{\text{root}})$. Each tree node has at most 2^d active links to the nodes of the higher level, where d is the dimension of the computational domain.

While traversing a tree structure during a node access, we are normally getting a cache miss each time we are following a link to a higher level node. In order to decrease the number of cache misses, and improve code performance, a cache-friendly storage for the tree nodes has been implemented. Starting from the finest level of resolution, nodes of each two levels are stored together; hence no cache miss occurs during the link following between these two levels.

This data structure has been developed for the serial implementation of the wavelet based code. The parallel version uses the same data structure without any changes, with each processor having its own tree-structure containing all tree roots. The question whether to keep some neighboring tree roots instead of all of the tree roots of the whole domain is currently under investigation. It might happen that the tree structure has to be adjusted for the parallel code requirements. However, taking into account that the storage overhead of having empty roots is insignificant, the algorithm simplicity and an ease of inter-processor communication are the good reasons to keep the current implementation.

Parallel communicators have been implemented via collective all-to-all MPI calls, which eliminated all the deadlock related problems and facilitated code debugging. From the other

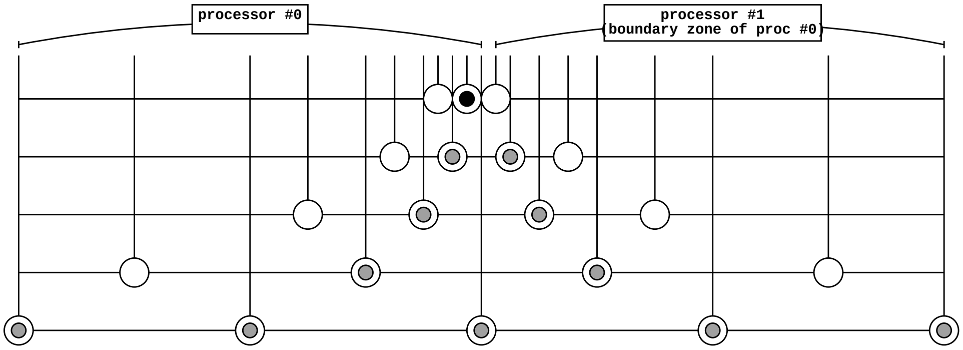


Figure 28: An example of 1-D boundary zone of processor 0 generated by a single active wavelet node (black dot) consists of nodes required for wavelet transform in that active node (gray dots) and nodes required to find the derivatives in that node (white circles) for $n_{\text{prdct}} = n_{\text{diff}} = 2$ and $n_{\text{updt}} = 0$. Thus, processor 0 has to transfer 5 points before wavelet transform, and at least 5 (up to 9, depending on the distribution of active wavelets on processor 1) points before derivative calculations.

hand, the major trade-off of such an approach is the performance of the code. Since a processor would require information from several neighboring domains only, not from all the other processors, replacing all-to-all with point-to-point communications has good chances to improve code performance (especially for a large number of processors).

Currently, all communicators are written to be independent from the quad-tree data structure and kept in one file for simplicity. Data structure independence could facilitate future development of the parallel code with different underlying data structures. Also for simplicity of development, throughout the code parallel related things are surrounded by preprocessor statements `#ifdef MULTIPROC`.

Numerous changes have been introduced into the main wavelet code as well as into the main code - data structure interface. In short, following the domain decomposition ideology, each processor controls wavelets located inside the processor sub-domain. In addition, each processor has a boundary zone around the sub-domain to store and exchange information between the processors. The origin of the boundary zone lays in wavelet transform and derivative calculations (Fig 28). In order to find derivatives in a point we need a prescribed number of points on each side of that point. Similarly, to perform a wavelet transform inside a sub-domain we need a predefined number of points on each side of each point where wavelet coefficient is calculated during the predict stage.

Having an update stage imposes tight coupling between the neighboring sub-domains and requires multiple information exchanges between the sub-domains (namely, at each level of the wavelet transform). This drastically decreases code performance and therefore to be avoided. For more detailed discussion of explored possible parallel extension of the algorithm see Section 6.2. In the current implementation we are not using update during wavelet transform, which results in a slight increase of the approximation errors. Though, wavelet threshold value has a stronger effect on the approximation errors, and a smaller threshold would easily compensate the absence of update during wavelet transform. Nevertheless, the

System	N	Time, s	Efficiency, %
1 node,	1	1215	100
128×Itanium2,	8	236	64
NUMA, shared	16	237	32
	27	267	17
1 node,	1	607	100
4×Xeon,	2	330	92
shared	3	278	73
	4	228	66
267 nodes,	1	762	100
4×Opteron,	4	261	73
Myrinet	8	180	53
	16	172	28

Table 4. Scalability results for Parallel Adaptive Wavelet Collocation Solver.

necessity of the parallel wavelet transform with both predict and update stages for some applications is currently under investigation.

All points required for wavelet transform are points inside the sub-domain (so called "significant" points of the current sub-domain) are added to the data structure and marked as "adjacent". All points from the derivatives stencils, as well as the points required to make wavelet transform in these points, are added to the data structure as "ghosts". By the end of the grid adaptation subroutine each processor has the access to these lists and, after synchronization, to the lists of the other processors. Thus, in all the following information exchanges on the current grid, e.g. during numerous wavelet transforms and derivative calculations, each processor will know which nodes it has to send and which nodes it is going to receive from other processors. Therefore, value synchronization during wavelet transform require a single all-to-all MPI communication to transfer "adjacent" nodes. The synchronization during derivative calculations require transfer of "adjacent" and "ghosts" nodes.

It should be noted that "ghosts" values, otherwise zeroes, are initialized during that inverse transform only. Value at some "ghost" point could be changed, and therefore need to be synchronized before the inverse transform, if and only if that "ghost" has participated in the previous direct forward transform, i.e. that "ghost" is "significant" or "adjacent" at some other processor. All the other "ghosts" are safely removed from the synchronization lists to decrease the communication cost.

In addition to grid adaptation and derivatives, solvers are the important parts of the main code to run in parallel. In the current implementation we extensively use MPI global

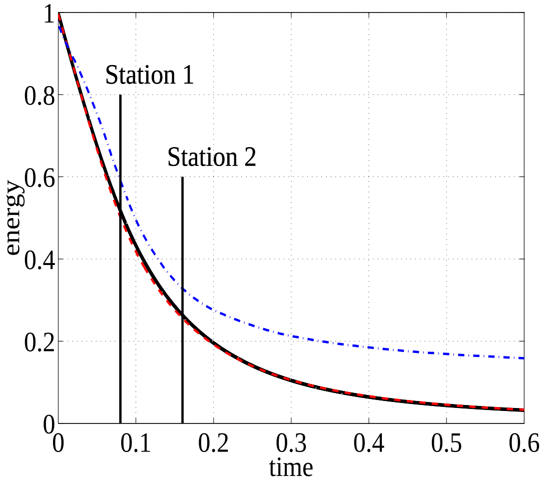


Figure 29: Energy decay for Case$_{\mathrm{Re}_\lambda=72}$ with partial adjacent zone for CVS with no SGS model (– – – –), SCALES with no SGS model (– · – · –) and for comparison DNS (———). Large eddy turn over time for the initial DNS field is approximately 0.1. Two stations are shown at which energy spectra will be presented.

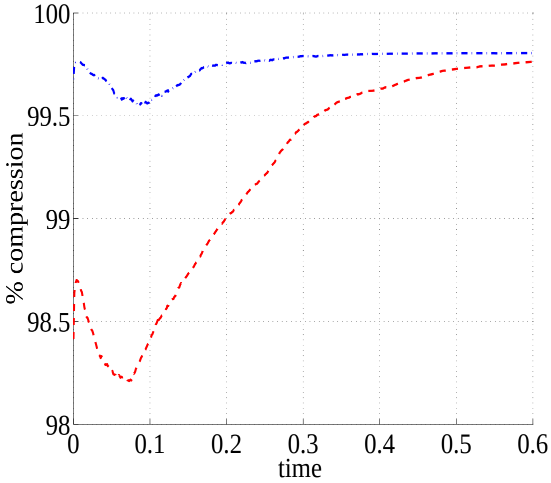


Figure 30: Field compression for Case$_{\mathrm{Re}_\lambda=72}$ with partial adjacent zone for CVS with no SGS model (– – – –) and SCALES with no SGS model (– · – · –).

reduction operations (mainly, summation) inside the solvers.

6.3.5 Current Performance and Future Work

Current parallel implementation has been tested on several systems: single node NUMA (non-uniform memory access) shared memory system with 128 Itanium2 processors, single shared memory node with 4 Xeon processors, and a cluster of 267 4×Opteron nodes interconnected by Myrinet 2g network. Timing results are presented in Table 4 for different number of processors N. Efficiency is measured relative to a perfect application which would run N times faster on N processors. In short, current parallel implementation of wavelet based code runs efficiently on up to 8 processors. Future work is required to improve the performance of the parallel code.

Future work includes the following (in order of importance): (1) Extensive profiling and bottleneck detection. (2) Tree structure tunning. (3) Replacing all-to-all with point-to-point communications. (4) Load balancing. (5) Additional optimization of parallel solvers is required. (6) Parallel wavelet transform with non-zero n_{update}.

7 Applications

7.1 Decaying Isotropic Turbulence

To validate the CVS and SCALES methods, numerical simulations of decaying incompressible isotropic turbulence are considered. For this project the incompressible Navier–Stokes

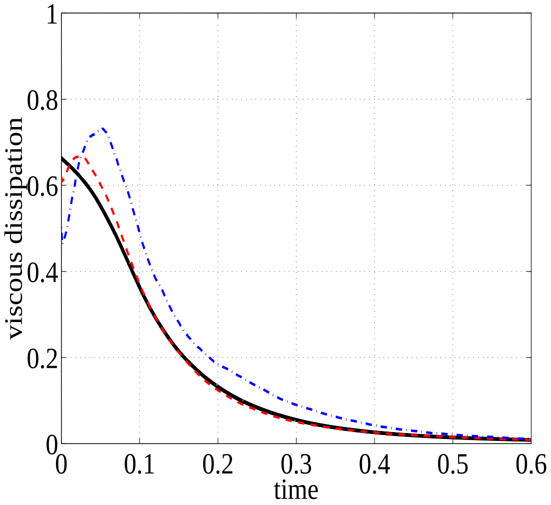


Figure 31: Total resolved viscous dissipation $(-\nu S_{ij} S_{ij})$ for $\mathrm{Case}_{\mathrm{Re}_\lambda=72}$ with partial adjacent zone for CVS with no SGS model (- - - -), SCALES with no SGS model (— · —) and for comparison DNS (——).

equations (11–13) are solved with the DAWC solver. Continuity (13) is enforced using a multi-step pressure correction time integration method (Guermond & Shen 2003). An adaptive wavelet collocation multilevel elliptic solver (Vasilyev & Kevlahan 2005) is used in solving the Poisson equation for pressure at each time step.

Results of decaying incompressible isotropic turbulence with initial $\mathrm{Re}_\lambda = 72$ are presented. The simulations were initialized with a 128^3 forced isotropic turbulence DNS field from a de-aliased pseudo-spectral code. The DNS simulation was run using a resolution of 128^3 and had an initial eddy-turnover time of approximately 0.1. The spectral content of the initial DNS field is fully resolved by doubling the non-adaptive field resolution to 256^3 in the simulations. This is required because the DAWC solver uses finite differences, which cannot resolve the full spectral content of the spectral DNS field at the original resolution. The results are compared to a full DNS performed with the de-aliased pseudo-spectral code used to generate the initial DNS field. Currently we are experimenting with larger runs with effective resolution 1024^3 for both CVS and SCALES simulations. What is remarkable that SCALES simulations with such high resolution use less than 2GB memory, which is very encouraging, since it points out to the great compression achieved by the algorithm due to high intermittency of turbulent flows.

In running these simulations it has been determined that a more "complete" adjacent zone, than the partial adjacent zone described in Section 2.5, is needed to limit the numerical and aliasing error at the high field compression used in SCALES. In the original adjacent zone, neighboring points on the level above, the current level, and the level below are added around each active wavelet. From this point on, we will refer to this as a partial adjacent zone. This partial adjacent zone is used in the CVS simulations presented in Section 2.5. For the high compression SCALES simulations presented in Section 7.3, 7.4, 7.4, 7.5, and

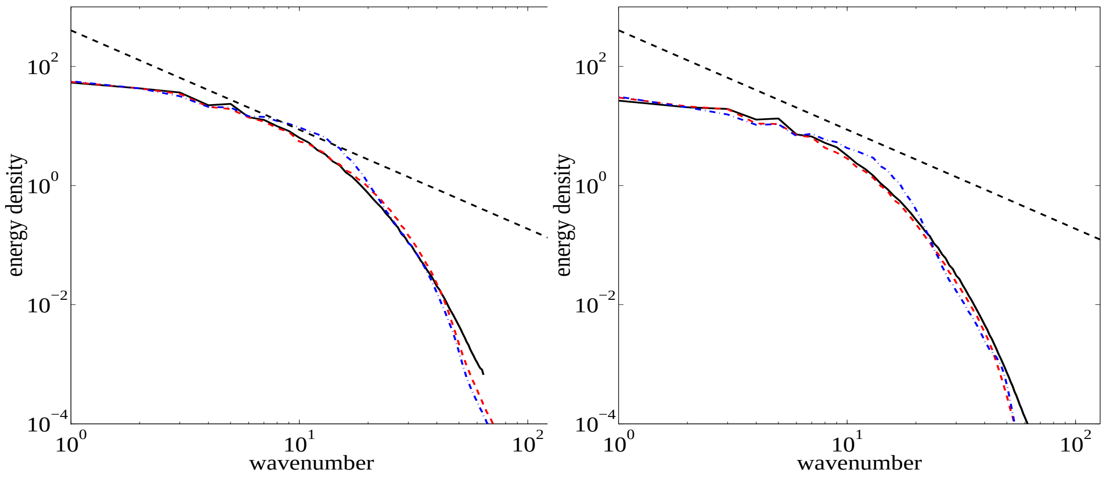


Figure 32: Energy spectra for Case$_{Re_\lambda=72}$ with partial adjacent zone for CVS with no SGS model (– – – –), SCALES with no SGS model (– – – –) and for comparison DNS (———), at time $t = 0.08$ (left), and $t = 0.16$ (right). A $k^{-5/3}$ straight dashed black line is shown to indicate the inertial range.

7.6 we have defined a complete adjacent zone that, in addition to the immediate neighbors, adds the diagonal neighbors.

For all SCALES and CVS results in this report the L_2 WTF-norm was used in the coupled wavelet filtering for grid adaptation based on the velocity field and for grid and test filtering in the model. Tests were also run using L_∞ WTF-norm but it was determined that this was considerably nosier due to temporal intermittency.

7.2 CVS

CVS simulations of decaying incompressible isotropic turbulence have been performed with no SGS stress model to validate the method's ability to dynamically resolve and track the coherent energetic eddies in a turbulent flow. A partial adjacent zone has been used for these simulations. In figures 29 through 32 the results of CVS and SCALES with no SGS stress model, for brevity called SCALES$_{\text{no}-\text{mdl}}$, are compared to DNS . It can be seen in Fig. 29 that the energy decay for CVS is nearly identical to the DNS. The SCALES$_{\text{no}-\text{mdl}}$ case is seen to be under dissipative. Figure 30 shows the field compression for CVS and SCALES$_{\text{no}-\text{mdl}}$. The compression stated is always with respect to the maximum field resolution, which in this case is 256^3. It can be seen that CVS is able to reproduce the DNS energy decay with a compression ranging from a minimum of 98.2% to 99.8% as the flow becomes laminar. This means that a maximum of 1.8% of the total modes are resolved in the CVS simulation. This variation of field compression over the course of the simulation reflects the decreasing amount of small scale structures as the turbulence intensity decreases. In these simulations ϵ is set to 0.15 for CVS and 0.5 for SCALES$_{\text{no}-\text{mdl}}$. The value of ϵ for CVS was chosen iteratively to find

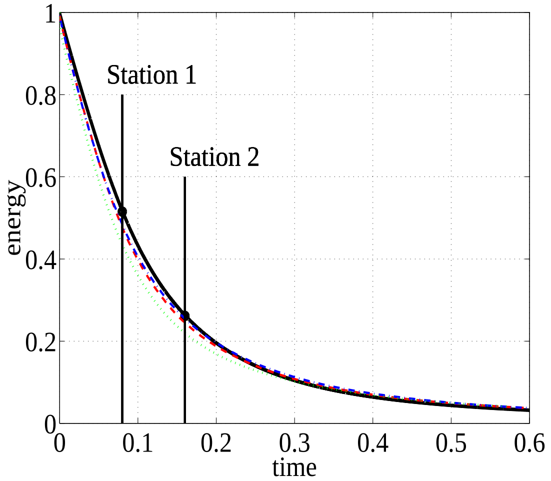


Figure 33: Energy decay for Case$_{\mathrm{Re}_\lambda=72}$ with complete adjacent zone for SCALES with dynamic SGS model (−−−−), SCALES with SGS model coefficient $C_\epsilon \ell^2 \epsilon^2 = 0.0001$ (−−−), LES with dynamic SGS model (—) and for comparison DNS (——). Large eddy turn over time for the initial DNS field is approximately 0.1 . Two stations are shown at which energy spectra will be presented.

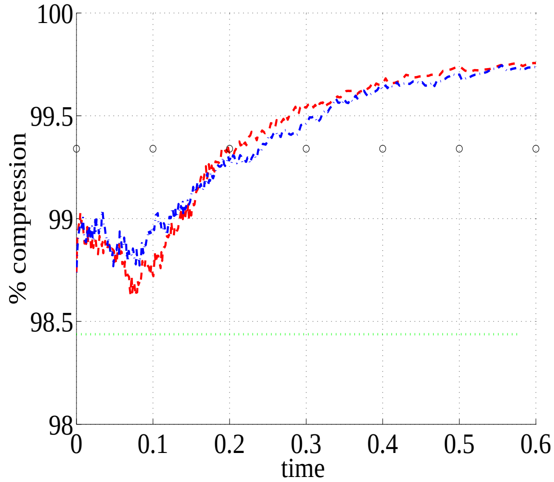


Figure 34: Field compression for Case$_{\mathrm{Re}_\lambda=72}$ with complete adjacent zone for SCALES with dynamic SGS model (−−−−), SCALES with SGS model coefficient $C_\epsilon \ell^2 \epsilon^2 = 0.0001$ (−−−) and LES with dynamic SGS model (—). The complete interpretation of the LES compression based on the 3/2 rule is shown as small circles.

the maximum value for which the energy decay over the simulation period closely matched that obtained with the DNS. For comparison CVS simulations (not shown) with $\mathrm{Re}_\lambda = 48$ have been run. For these $\mathrm{Re}_\lambda = 48$ simulations 6% of the modes were required to match the DNS energy decay. This trend of compression scaling between $\mathrm{Re}_\lambda = 48$ and $\mathrm{Re}_\lambda = 72$ is believed to be indicative of the expected scaling of CVS compression with Reynolds number. More data points are needed at higher Reynolds number to validate this possible level of scaling. In this CVS simulation the skewness of the first velocity derivative is maintained to within 10% of the DNS value, which reflects the fact that the CVS is resolving most of the DNS energy dissipation. In Fig. 31 we see directly that, after an initial period where the small scales are being recovered from the initial field projection, the total viscous dissipation of CVS closely matches the DNS. This confirms the hypothesis that with CVS the total SGS dissipation is minimal. This also indicates that the CVS is capturing the coherent structures, allowing the CVS simulation to at least partially resolve the energy cascade over all active wavenumbers. In Fig. 32 the energy spectra for CVS, SCALES$_{\mathrm{no-mdl}}$ and DNS are shown for two stations. The first station is at $t = 0.08$ and the second station is at $t = 0.16$. These stations are also shown on Fig. 29. The CVS spectra closely matches that of the unfiltered DNS at both stations. Notice how with CVS the full energy spectra is closely resolved over the full spectral range. The spectra for the SCALES$_{\mathrm{no-mdl}}$ case is seen to build up energy due to lack of SGS dissipation.

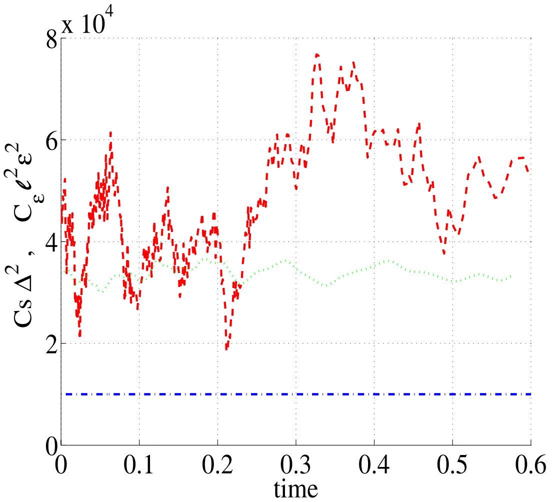


Figure 35: Dynamic SGS model coefficient for $\text{Case}_{\text{Re}_\lambda=72}$ with complete adjacent zone for SCALES with dynamic SGS model, $C_\epsilon \ell^2 \epsilon^2$ (- - - -), SCALES with SGS model coefficient $C_\epsilon \ell^2 \epsilon^2 = 0.0001$ (—·—) and LES with a classical dynamic SGS model, $C_s \Delta^2$(······).

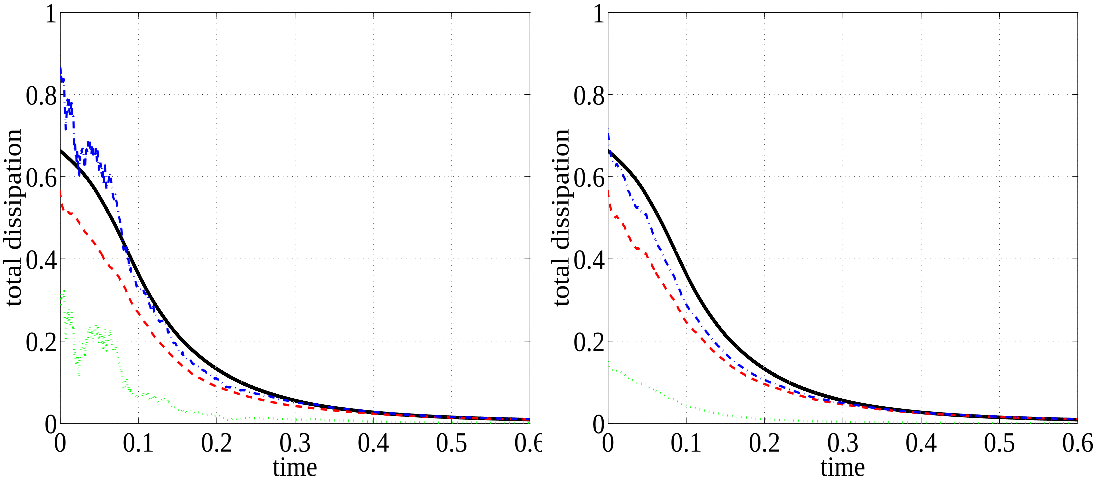


Figure 36: Resolved and SGS dissipation for SCALES with the dynamic model (left) and SCALES with SGS model coefficient $C_\epsilon \ell^2 \epsilon^2 = 0.0001$ (right) for $\text{Case}_{\text{Re}_\lambda=72}$ with complete adjacent zone. On both plots the DNS viscous dissipation is shown (——)with the viscous dissipation (- - - -), SGS dissipation (······) and viscous + SGS dissipation (—·—).

7.3 SCALES Constant Coefficient and Global Dynamic SGS Model

SCALES simulations have been performed with the constant coefficient Smagorinsky eddy viscosity model, equation (16), and the new dynamic Smagorinsky eddy viscosity SGS stress model described in Section 4. The model coefficient ($C_\epsilon \ell^2 \epsilon^2 = 0.0001$) for the $\text{SCALES}_{\text{Cs}}$ case was chosen to best match the DNS results. For the $\text{SCALES}_{\text{dyn}}$ case the volume averaged version of the dynamic model coefficient is used (27). These SCALES simulations, hereafter for brevity called $\text{SCALES}_{\text{Cs}}$ and $\text{SCALES}_{\text{dyn}}$ respectively, are compared to DNS and LES simulations. For both $\text{SCALES}_{\text{Cs}}$ and $\text{SCALES}_{\text{dyn}}$ cases ϵ is set to 0.5. The LES simula-

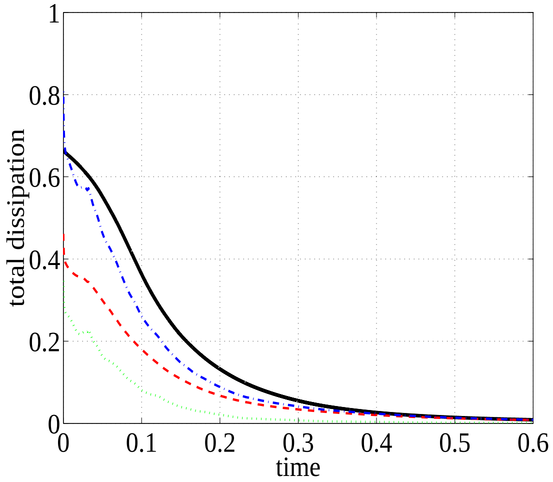


Figure 37: Resolved and SGS dissipation for LES with the dynamic model for $\mathrm{Case}_{\mathrm{Re}_\lambda=72}$. The DNS viscous dissipation is shown (——) with the viscous dissipation (- - -), SGS dissipation (—) and viscous + SGS dissipation (-·-·).

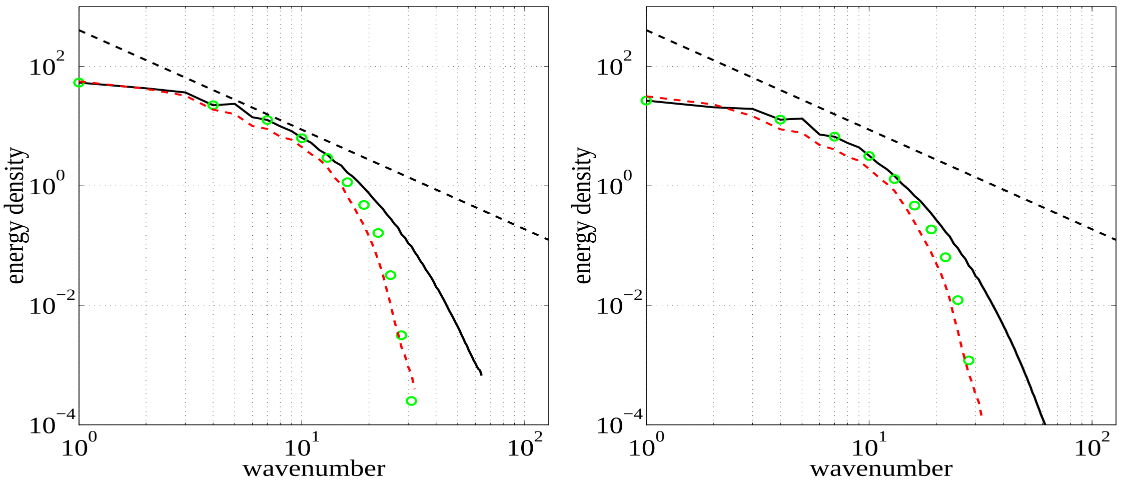


Figure 38: Energy spectra for $\mathrm{Case}_{\mathrm{Re}_\lambda=72}$ with complete adjacent zone for LES with dynamic SGS model (- - -) at time $t = 0.08$ (left), and $t = 0.16$ (right). For comparison the DNS (——) and filtered DNS (○) are shown. A $k^{-5/3}$ straight dashed black line is shown to indicate the inertial range.

tion is performed in the DAWC solver with a regular 64^3 grid using the classical dynamic Smagorinsky model. The simulation is de-aliased by performing a wavelet transform on the velocity field and zeroing the highest level wavelet coefficients, thus resulting in a 32^3 solution at the end of the time step. This is more expensive than the 3/2 rule used in pseudo-spectral simulations. Figure 33 shows that the resolved kinetic energy decay for the $\mathrm{SCALES}_{\mathrm{dyn}}$ and $\mathrm{SCALES}_{\mathrm{Cs}}$ cases closely matches that of the DNS. The LES deviates slightly more from the DNS. Note that due to the similarity of the $\mathrm{SCALES}_{\mathrm{dyn}}$ and $\mathrm{SCALES}_{\mathrm{Cs}}$ results the lines are difficult to distinguish on Fig. 33. The $\mathrm{SCALES}_{\mathrm{dyn}}$ and $\mathrm{SCALES}_{\mathrm{Cs}}$ results lines are those just below the DNS line and above the LES line. In Fig. 34 the compression for the $\mathrm{SCALES}_{\mathrm{dyn}}$,

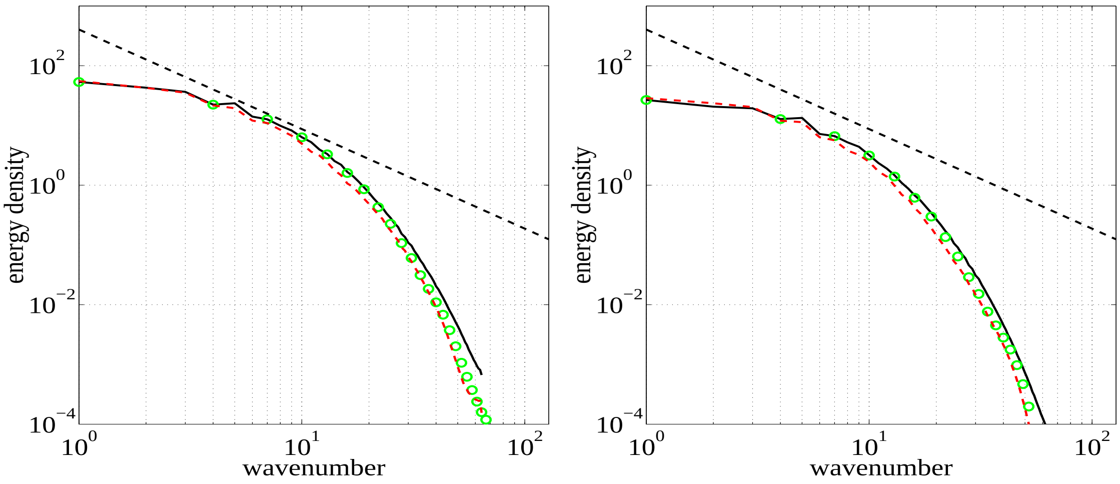


Figure 39: Energy spectra for Case$_{\mathrm{Re}_\lambda=72}$ with complete adjacent zone for SCALES with SGS model coefficient $C_\epsilon \ell^2 \epsilon^2 = 0.0001$ (– – – –) at time $t = 0.08$ (left), and $t = 0.16$ (right). For comparison the DNS (——) and filtered DNS (○) are shown. A $k^{-5/3}$ straight dashed black line is shown to indicate the inertial range.

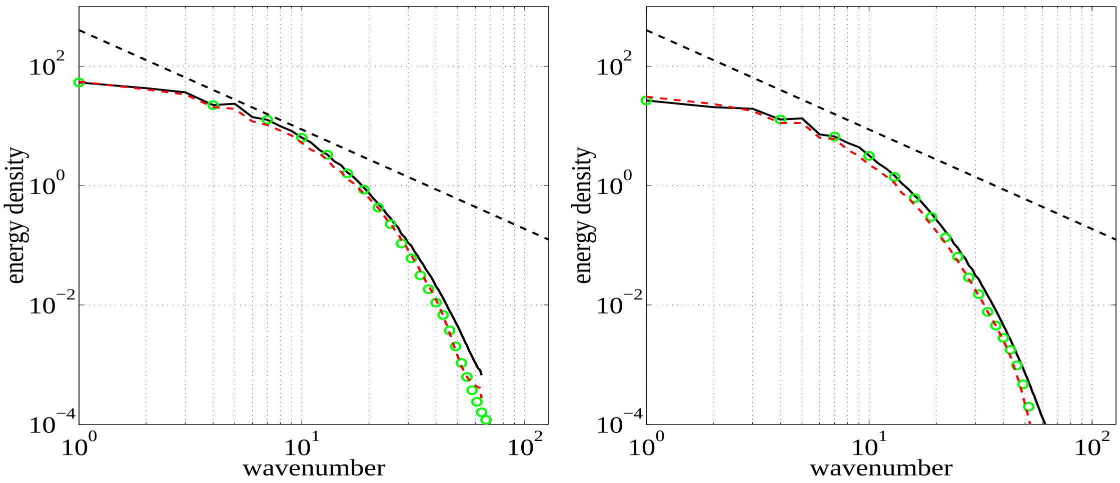


Figure 40: Energy spectra for Case$_{\mathrm{Re}_\lambda=72}$ with complete adjacent zone for SCALES with dynamic SGS model (– – – –) at time $t = 0.08$ (left), and $t = 0.16$ (right). For comparison the DNS (——) and filtered DNS (○) are shown. A $k^{-5/3}$ straight dashed black line is shown to indicate the inertial range.

SCALES$_{\mathrm{Cs}}$ and LES cases are shown. If we consider the overhead of the modes used for de-aliasing, the LES may be considered to have a compression of 98.44%. The modes used for de-aliasing in LES can be considered as analogous to the adjacent zone in SCALES, so for a realistic comparison we can consider that if the LES was performed in a spectral code, using the 3/2 rule for de-aliasing, the effective compression would be 99.34% (shown in Fig. 34 as small circles). This is 0.35% higher than the initial compression of the SCALES$_{\mathrm{dyn}}$ simulation. However, as the SCALES simulations progress the adaptive compression increases, surpassing that of the LES. Therefore, it can be said that the SCALES$_{\mathrm{dyn}}$ and SCALES$_{\mathrm{Cs}}$

simulations were able to capture the energy decay with a compression similar to a de-aliased LES simulation. In Fig. 35 we see that the dynamic model coefficient for SCALES$_{\mathrm{dyn}}$ is more variable in comparison to the LES case. We conjecture that this variability could reflect the sensitivity of the SCALES$_{\mathrm{dyn}}$ model to actual localized events, such as energetic coherent vortex interactions that cause local high resolved stresses. These events must be included to properly characterize the instantaneous SGS dissipation. Further research is needed to understand this phenomenon. In Figs. 36 and 37 the viscous and SGS dissipations are presented for SCALES$_{\mathrm{dyn}}$, SCALES$_{\mathrm{Cs}}$ and LES. The variability of the SCALES$_{\mathrm{dyn}}$ model coefficient is reflected in the SCALES$_{\mathrm{dyn}}$ SGS dissipation. In Figs. 38-40 the energy spectra for the two stations shown in Fig. 33 are compared to the appropriately filtered DNS for the SCALES$_{\mathrm{dyn}}$, SCALES$_{\mathrm{Cs}}$ and LES cases. For comparison to LES the DNS is filtered using a spherical Fourier cutoff filter equivalent to the maximum wave number resolved in the LES calculation. In the case of the SCALES simulations the appropriate DNS filtering for comparison is a wavelet thresholding filter with equivalent threshold parameter to that used in the SCALES simulations. It can be seen that, while there is reasonable agreement for the LES case (Fig. 38), the agreement with the filtered DNS is significantly improved for the SCALES$_{\mathrm{Cs}}$ (Fig. 39) and SCALES$_{\mathrm{dyn}}$ cases (Fig. 40). At both stations, in the dissipative range, the SCALES$_{\mathrm{Cs}}$ and SCALES$_{\mathrm{dyn}}$ simulations reproduce more of the high wave number energy. At the second station it can be seen in the inertial range that the LES has dissipated slightly more then the SCALES$_{\mathrm{Cs}}$ and SCALES$_{\mathrm{dyn}}$ cases. It is of particular interest to note that the wavelet filtered DNS in figures 39 and 40 are closer to the full DNS spectra over the full spectral range. Thus, the ability of SCALES to closely recover the filtered DNS results in a solution that has a spectral content close to the original unfiltered DNS solution over the whole DNS spectral range.

7.4 Lagrangian Dynamic Local SGS Model

In this section we apply the local Lagrangian SCALES model to incompressible isotropic decaying turbulence. Though the localized models are specifically designed to simulate complex inhomogeneous turbulent flows, it is nevertheless enlightening to test them for a case for which well known theoretical and experimental results exist. In addition, the homogeneous turbulence case allows a detailed comparison with the existing reference solutions for DNS, LES, and SCALES (with global dynamic model). Moreover, decaying turbulence is a challenging example of statistically unsteady flow that stands as a good test-case for *a-posteriori* verifying the accuracy of the proposed SGS stress model. The initial velocity field is a realization of a statistically stationary turbulent flow at $Re_\lambda = 72$ (λ being the Taylor microscale) that is provided by a fully de-aliased pseudo-spectral DNS with 128^3 Fourier modes (De Stefano *et al.* 2005). Due to the finite difference nature of the AWCM solver, the initial SCALES resolution in each direction must be doubled in order to retain the spectral

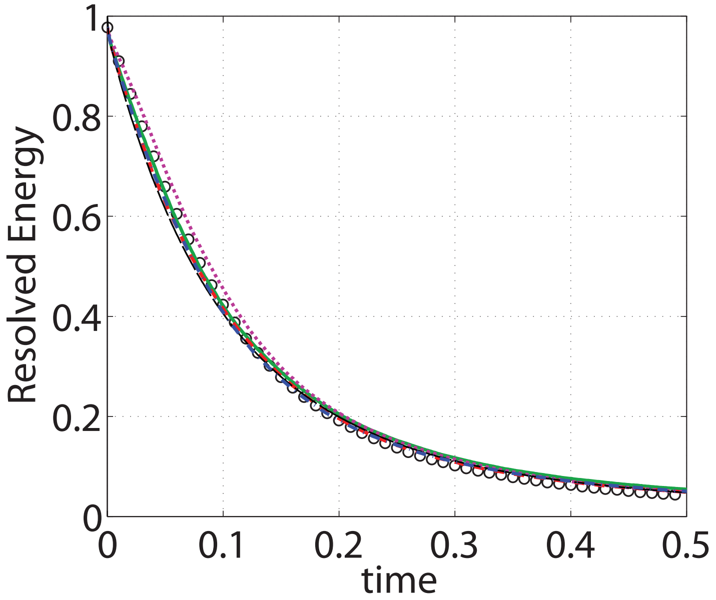


Figure 41: Energy decay for SCALES with the Lagrangian path-line diffusive (– – – –) and path-tube averaging (– — – — –) models, SCALES with global dynamic model (———), SCALES with no model (···), reference LES with global dynamic model (— —), and wavelet filtered DNS (○).

energy content. In other words, SCALES is run using a maximum resolution corresponding to 256^3 grid points. Note that due to the nature of the decaying turbulence, 256^3 resolution is only required during initial times with gradual decrease of the maximum level of resolution as turbulence kinetic energy decays and Taylor microscale Reynolds number decreases.

The choice of the thresholding parameter, ϵ, in (5) is somewhat arbitrary: the smaller it is, the weaker the SGS dissipation is, with SCALES approaching Coherent Vortex Simulation (Farge *et al.* 1999, Goldstein *et al.* 2005) and wavelet-based DNS for even smaller values of $\epsilon \leq 10^{-3}$. On the other hand, when ϵ is too large, too many modes are discarded and the energy cascade is no longer captured. All the SCALES results reported in this report have been obtained using the wavelet thresholding parameter $\epsilon = 0.43$ as a compromise between these limits.

The Lagrangian local modelling variables are initialized as $\mathcal{I}_{MM} = M_{hk}M_{hk}$ and $\mathcal{I}_{LM} = \bar{C}_s\epsilon^2\mathcal{I}_{MM}$, where $\bar{C}_s$ is the volume averaged Smagorinsky model coefficient (34). For the relaxation time scale, the value $\theta = 0.75$ suggested by (Meneveau *et al.* 1996) is chosen. For Lagrangian path-line diffusive averaging, a diffusion coefficient $C_{\mathcal{I}} \geq 5$ has been found to produce acceptable results while approaching the global dynamic model for very large values of the coefficient. In the case of Lagrangian path-tube averaging, the same stabilizing effect is obtained by means of local volume averaging along the path-line, provided that the linear cross-sectional dimensions of the path-tube are not smaller than the local characteristic filter width, Δ.

In figures 41–43 the resolved kinetic energy decay, grid compression (percentage of the

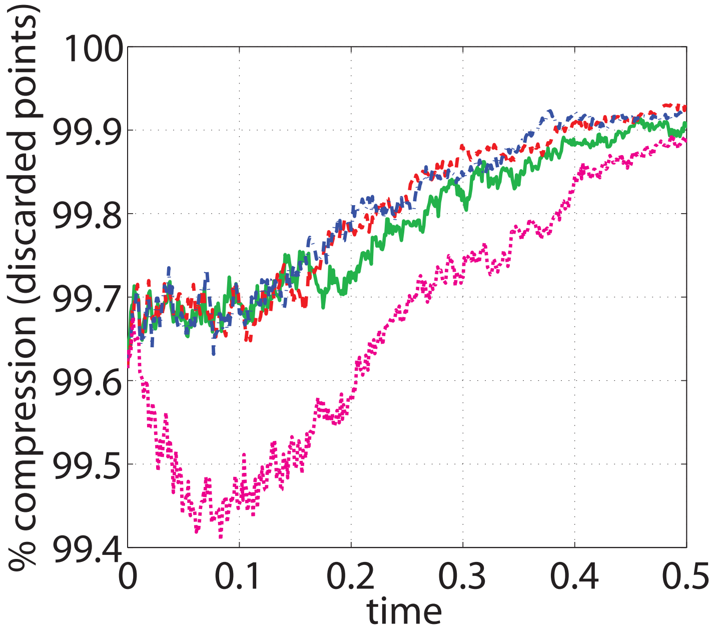


Figure 42: Field compression: SCALES with the Lagrangian path-line diffusive (- - - -) and path-tube averaging (— — — —) models, SCALES with global dynamic model (——), and SCALES with no model (—   ).

discarded grid points), and modelled SGS dissipation (percentage of the total dissipation) for proposed SCALES are compared to: *a-posteriori* wavelet filtered DNS, classical LES, SCALES with global dynamic model, and SCALES with no model. The LES is performed using the non-adaptive wavelet collocation solver on a regular 64^3 grid. The solution is de-aliased by performing a wavelet transform on the velocity field and zeroing the highest level wavelet coefficients for each time step. As to figure 41, the resolved kinetic energy is normalized with respect to the initial unfiltered DNS energy content. The grid compression is evaluated with respect to the maximum field resolution. The time scale used to report the results corresponds to approximately ten initial eddy-turnover times.

As can be seen from the energy decay plot in figure 41, the case of SCALES with no model is only slightly under-dissipative. From figure 44 we see that this case accurately captures the energy and enstrophy spectra. By examining the grid compression reported in figure 42 we can see that due to the adaptive nature of the numerical algorithm, the absence of SGS dissipation results in energy transfer to the small scales, where the energy is dissipated by viscous stresses. This process results in an increase in the number of the degrees of freedom and causes the solution to approach CVS (Goldstein *et al.* 2005). This effect would be more pronounced for higher Reynolds number flows, since SCALES with no model would have to resolve all the scales up to Kolmogorov scale and would rely on molecular viscosity as the only dissipative mechanism. The adaptive nature of the AWCM makes comparison of simulations tricky because the algorithm itself attempts to add resolution if the physical problem is under resolved. In this case good results are obtained at the cost of increased resolution. Another interesting observation is that despite the similar initial compression,

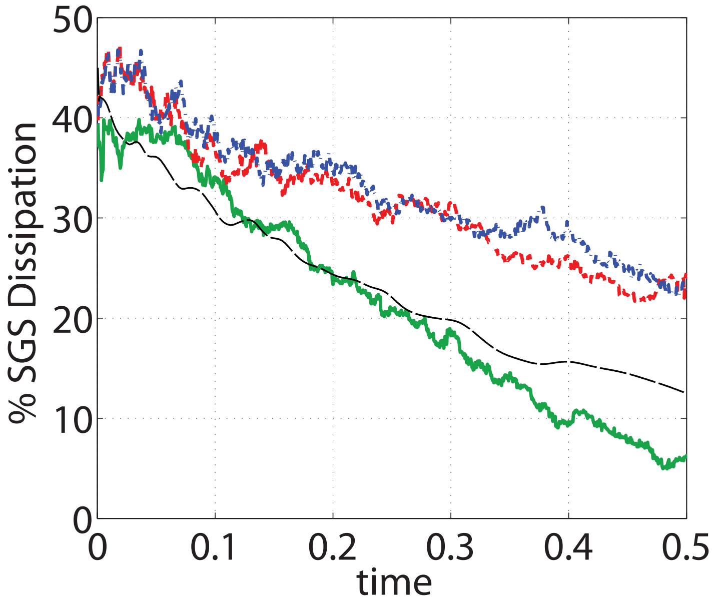


Figure 43: Percentage of SGS (modelled) dissipation: SCALES with the Lagrangian path-line diffusive (– – – –) and path-tube averaging (– – – – – –) models, SCALES with global dynamic model (———), and reference LES with global dynamic model (— —).

the local Lagrangian models show a higher level of SGS dissipation because they capture the local structure of the flow, rather than simply providing the necessary mean energy dissipation (as in the case of both the global dynamic model and LES). This, in turn, results in higher compression for larger times. It worth noting that the level of SGS dissipation of SCALES closely matches that of the global model and reference LES.

Another crucial feature of the SCALES approach is seen in the energy and, more importantly, enstrophy spectra, which are shown respectively in figures 44 and 45 for two different times. In contrast to classical LES, the SCALES results match not only in terms of temporal evolution of the total resolved turbulent kinetic energy, but also in terms of the DNS energy and enstrophy density spectra up to the dissipative wavenumber range. It is important to emphasize that this close match is achieved using less than 0.4% of the total non-adaptive nodes required for a DNS with the same wavelet solver. To highlight the significance of such a close match, it is interesting to compare these results with those of an LES with the global dynamic Smagorinsky model. Despite the fact that LES uses almost four times the number of modes (1.56%), it fails to capture the small-scale features of the spectrum. In addition, the total resolved LES kinetic energy is noticeably below the filtered DNS curve for moderate and high wavenumbers. These differences are even more pronounced for the enstrophy spectra.

It is worth stressing the unique feature of the SCALES approach, nameley the coupling of modelled SGS dissipation to grid compression: more grid points are used for models with lower levels of SGS dissipation. In other words, the SCALES approach compensates

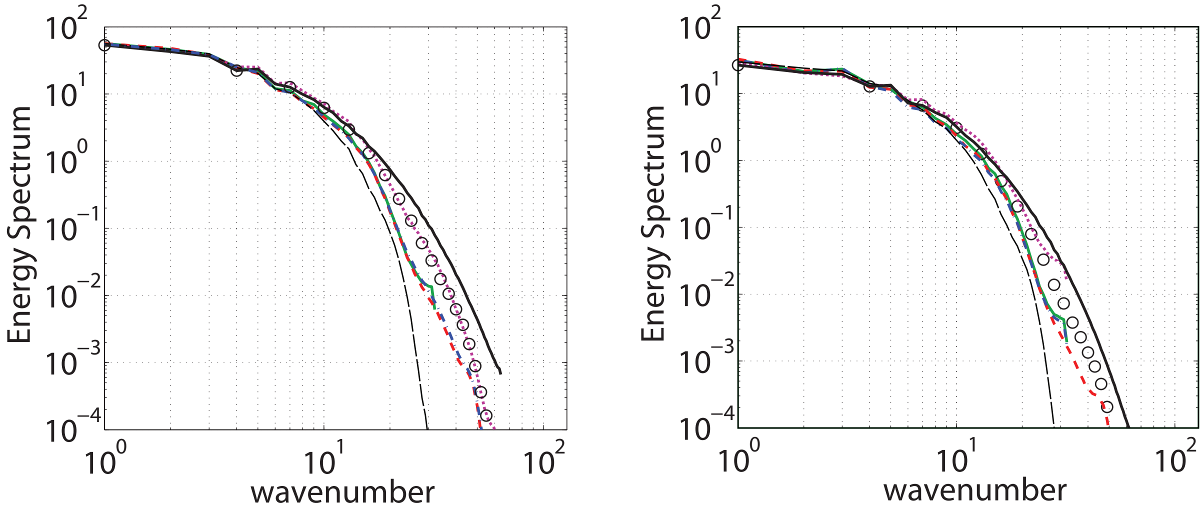


Figure 44: Energy density spectra at $t = 0.08$ (left) and $t = 0.16$ (right): SCALES with the Lagrangian path-line diffusive (- - - -) and path-tube averaging (— — — —) models, SCALES with global dynamic model (———), SCALES with no model (——), spectral DNS (▬▬), wavelet filtered DNS (○), and reference LES with global dynamic model (—— ——).

for inadequate SGS dissipation by increasing the local resolution and, hence, the level of resolved viscous dissipation. This can be seen clearly by comparing to the SCALES results with no SGS model.

As stated earlier, the absence of SGS dissipation results in the transfer of energy from low to high wavenumbers, filling the entire wavenumber range, bringing the energy and enstrophy spectra close to the wavelet filtered DNS spectrum. This processes continues until the lack of SGS dissipation is balanced by the viscous dissipation. Recall that enstrophy and viscous dissipation spectra are identical if properly normalized. Thus, an accurate enstrophy spectrum ensures proper viscous dissipation. On the other hand, the increase of energy in high wavenumber range results in an increase of degrees of freedom (active wavelet coefficients or grid points), as seen in figure 42. The energy and enstrophy spectra for SCALES with local Lagrangian models closely match each other and agree reasonably well with the spectra for filtered DNS. The non-local character of the dissipation of the global dynamic model results in over-dissipation at small scales and, subsequently, smaller wavelet coefficients on the finest level of resolution, which ultimately results in the earlier removal of the finest level of resolution from the adaptive computational grid.

Finally, SCALES with the local dynamic Smagorinsky model with both types of Lagrangian averaging are virtually identical, which highlights the similarities of both averaging approaches.

7.5 Kinetic Energy Based SGS Models

In this section, the results of the numerical experiments are presented and discussed. The proposed one-equation models, summarized for the sake of clarity in Table 1, are evaluated by performing SCALES of incompressible isotropic freely decaying turbulence in a cubic box

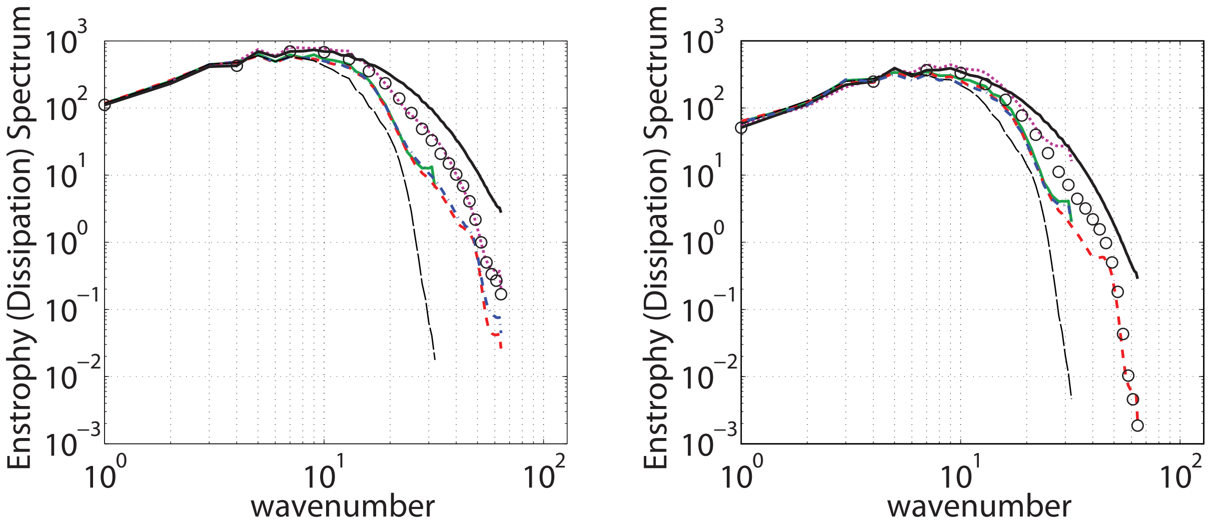


Figure 45: Enstrophy density spectra at $t = 0.08$ (left) and $t = 0.16$ (right): SCALES with the Lagrangian path-line diffusive (- - - -) and path-tube averaging (– – – – –) models, SCALES with global dynamic model (——), SCALES with no model (—), spectral DNS (━━), wavelet filtered DNS (○), and reference LES with global dynamic model (— —).

with periodic boundary conditions. Though these localized models are specifically designed to simulate complex non-homogeneous turbulent flows, it is nevertheless enlightening to test them for a case for which well known theoretical and experimental results exist. Moreover, decaying turbulence is a challenging example of statistically unsteady flow and is a good test-case for *a-posteriori* verifying the accuracy of both the SGS stress and the SGS energy dissipation models.

The simulation settings are chosen as follows. The initial velocity field is a realization of a statistically steady turbulent flow at $Re_\lambda \cong 72$ (λ being the Taylor microscale) as provided by a fully de-aliased pseudo-spectral DNS solution with 128^3 Fourier modes (De Stefano *et al.* 2005). The simulation of decaying turbulence is conducted for a temporal range of approximately ten initial eddy-turnover times that corresponds to a final value of $Re_\lambda \cong 22$. The relative threshold for wavelet-filtering (5) is set to $\epsilon = 0.43$ as in Lagrangian dynamic model. Due to the finite-difference nature of the AWCM solver, the initial resolution has been doubled in each direction in order to keep the DNS spectral energy content intact. In other words, SCALES is run using a maximum resolution corresponding to 256^3 grid points that corresponds to have $j_{\max} = 8$ in Eq.(5).

As regards the energy equation, the following initial condition has been used for the SGS kinetic energy: $k_{\text{sgs}}(\boldsymbol{x}, 0) = \beta_k \frac{\langle k^0_{\text{res}} \rangle}{\langle k^0_{\text{rts}} \rangle} k^0_{\text{rts}}$, where k^0_{res} and k^0_{rts} are evaluated, according to definitions (15) and (65), upon the initial wavelet filtered DNS field (Vasilyev *et al.* 2006). The coefficient β_k determines the initial ratio between residual and resolved energy that corresponds to the desired turbulence resolution. Based upon previous *a-priori* studies, it is set for the present experiments to $\beta_k = 0.1$.

In Fig. 46, the kinetic energy decay for the different models is illustrated, along with the reference GDM and wavelet-filtered DNS solutions. All the new proposed models capture

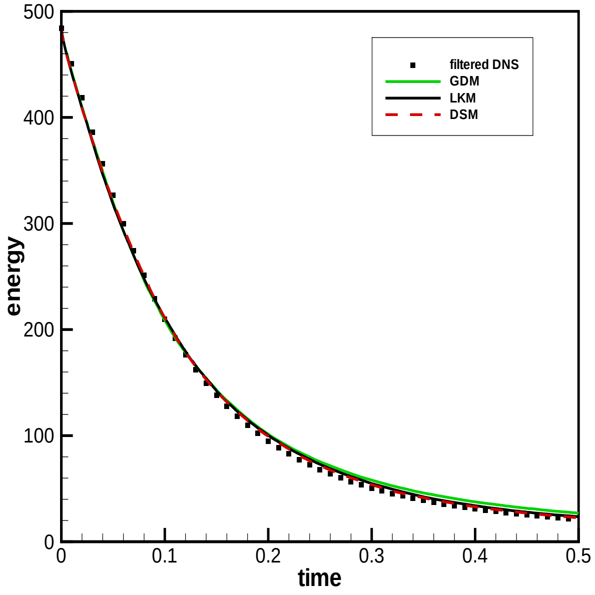


Figure 46: Energy decay. Left side: DSM (dashed red line) and LKM (———). Right side: LDKM-B (———) and LDKM-G (dashed red). The reference GDM (green line) and wavelet-filtered DNS (□) solutions are shown for comparison on both sides.

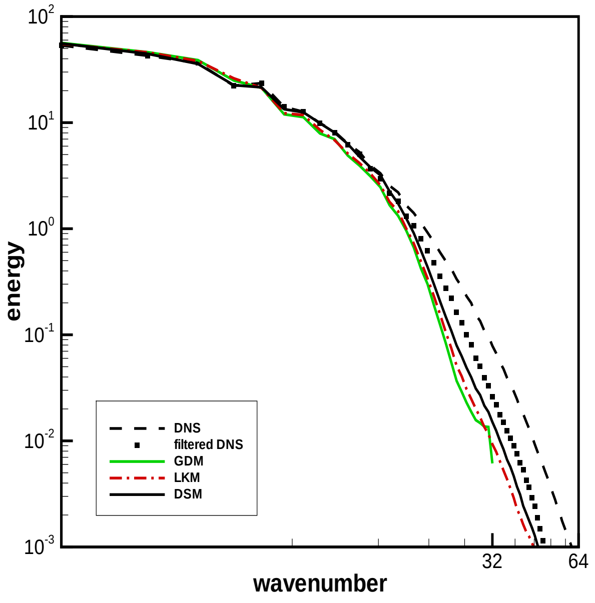


Figure 47: Energy spectra at $t = 0.08$. Left side: DSM (———) and LKM (red dashdotted line). Right side: LDKM-B (red dashdotted) and LDKM-G (———). The reference GDM (green line) and wavelet-filtered DNS (□) solutions are shown for comparison on both sides, along with the unfiltered DNS (– – – –).

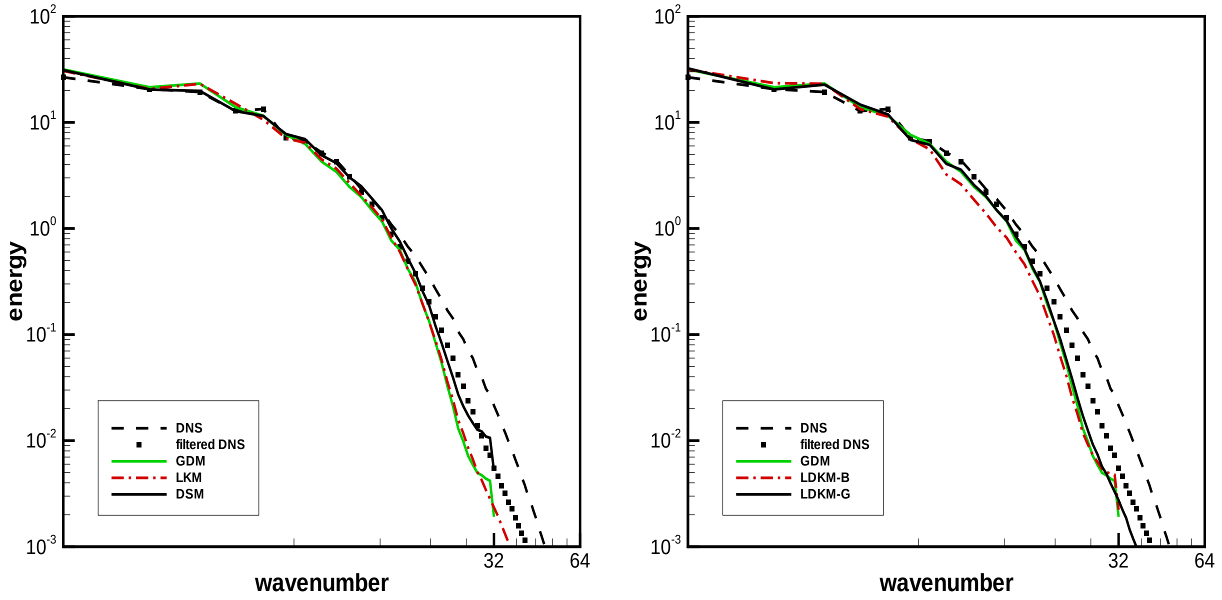


Figure 48: Energy spectra at $t = 0.16$. Left side: DSM (———) and LKM (red dashdotted line). Right side: LDKM-B (red dashdotted) and LDKM-G (———). The reference GDM (green line) and wavelet-filtered DNS (□) solutions are shown for comparison on both sides, along with the unfiltered DNS (- - - -).

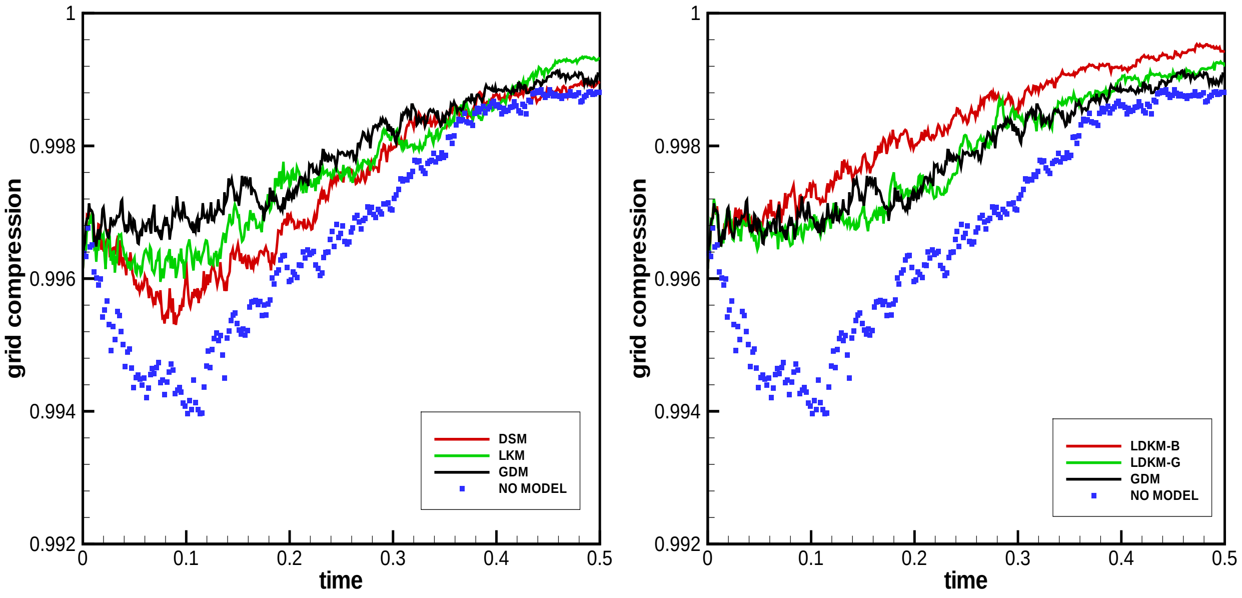


Figure 49: Grid compression. Left side: DSM (red line) and LKM (green). Right side: LDKM-B (red) and LDKM-G (green). The no-model (□) and GDM (———) solutions are shown for comparison on both sides.

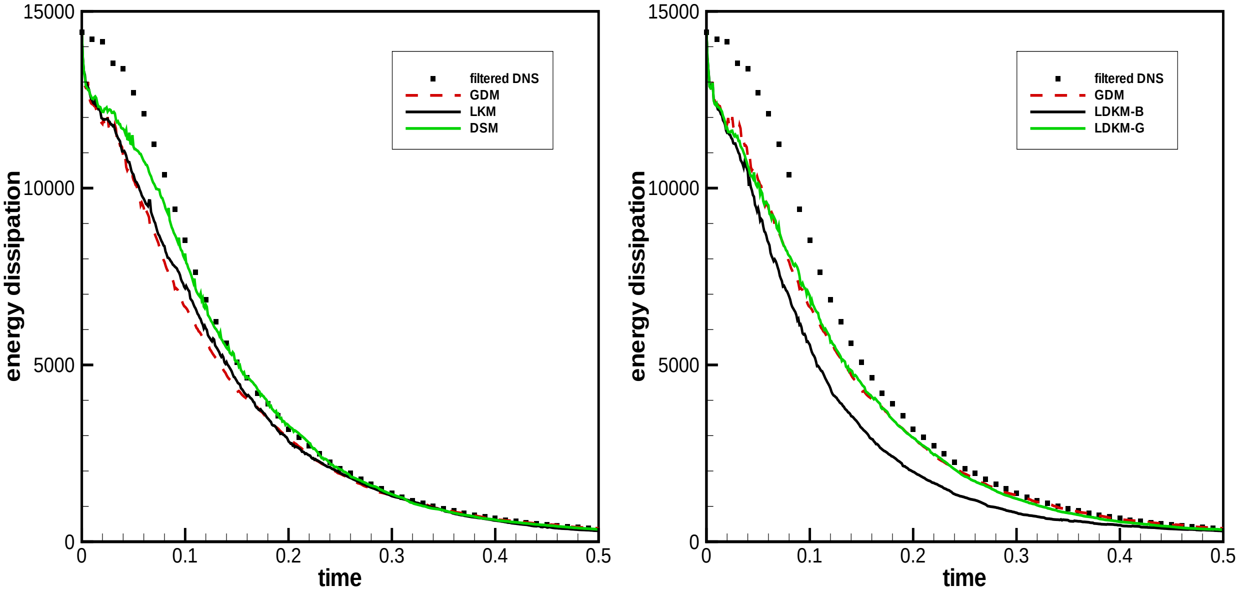


Figure 50: Energy dissipation. Left side: DSM (green line) and LKM (———). Right side: LDKM-B (———) and LDKM-G (green). The reference GDM (red dashed) and wavelet-filtered DNS (□) solutions are shown for comparison on both sides.

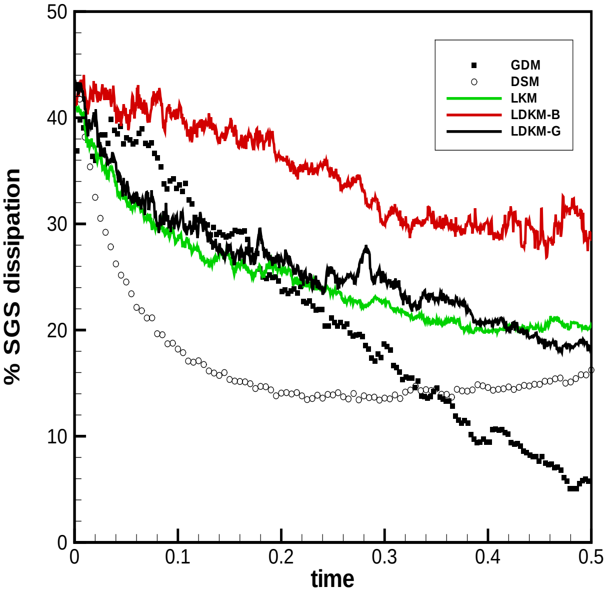


Figure 51: Percentage of SGS dissipation for the DSM (○), LKM (green line), LDKM-B (red) and LDKM-G (———) solutions, compared to GDM (□).

the energy decay slightly better than the global model. As to energy spectral distribution, Figs. 47 and 48 show the spectra at two different time instants or, equivalently, two different Re_λ, namely, $Re_\lambda \cong 46$ ($t = 0.08$) and $Re_\lambda \cong 35$ ($t = 0.16$). The localized dynamic SCALES solutions generally show acceptable energy spectra when compared to wavelet-filtered DNS at different times. Note that for the cases where there are no significant wavelet coefficients above level $j = 6$, the energy spectra lines stop at wavenumber 32.

Before going on with the discussion of the results, it is worth stressing the fact that modeled solutions showing the right energy decay as well as the correct energy spectra is not sufficient by itself to assess the effectiveness of the modeling procedure. In fact, the AWCM solver used in SCALES allows automatic refinement of the numerical mesh in flow regions where the model does not provide the adequate dissipation. For this reason, a deeper insight must be gained by examining the actual grid compression. As mentioned in Section 2.4, the compression can be defined as the ratio between the number of discarded and total allowable wavelets coefficients (or, equivalently, the same ratio in terms of grid-points). In order for the SCALES approach to be successful, the number of grid-points actually used during the simulation must be less than that required for a CVS solution of the same problem with no model. Otherwise, the adoption of a SGS model would appear useless, if not inappropriate.

The effectiveness of the SGS modeling is first demonstrated by making a comparison with the no-model solution. The latter has been found to be initially under-dissipative (see the following Fig. 52), thus confirming the need for the extra dissipation provided by the SGS model. However, the absence of modeled SGS dissipation results in energy transfer to the small scales, where the energy is dissipated by viscous stresses. Owing to the self-adaptive nature of the numerical method, this process results in increasing the number of resolved modes that causes the solution in practice to evolve towards the DNS approach.

As shown in Fig. 49, the gain in terms of compression with respect to the no-model solution is clear. The present grid compression is above 99.5% for all the different proposed models at all time instants, which corresponds to retaining about 1% of the 192^3 modes used for de-aliasing by the pseudo-spectral DNS (De Stefano *et al.* 2005). The achieved compression is comparable to the reference global dynamic model (Goldstein *et al.* 2005). The fact that different models show different compression, though using the same relative wavelet thresholding level, is not surprising because the adaptive-gridding is closely coupled to the flow physics and, therefore, it is strongly affected by the presence and type of the SGS stress model forcing.

The direct coupling of grid compression with resolved and SGS dissipation can be clearly seen by examining the corresponding Figs. 49, 50 and 51. The decrease of SGS dissipation (in the DSM case) results in the decrease of grid compression and the increase of resolved energy dissipation. That reenforces the above discussion about the effectiveness of the model. Also note that, despite the initial similar compression and similar initial level of SGS dissipation, the compression for the global dynamic model is higher. In fact, the non-local character

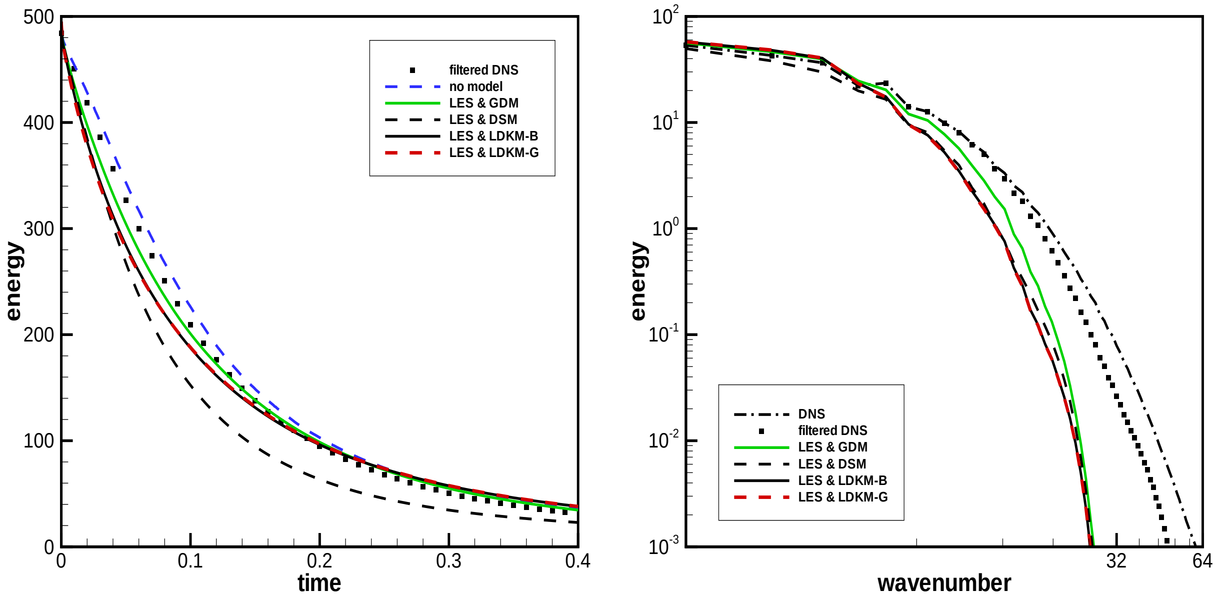


Figure 52: Energy decay (left) and energy spectra at t = 0.08 (right) for non-adaptive LES supplied with energy-based models: DSM (‒ ‒ ‒ ‒), LDKM-B (———) and LDKM-G (red dashed line), along with the corresponding GDM (green) and wavelet-filtered DNS (■) solutions. The unfiltered DNS (—·—) spectrum is shown for comparison.

of the GDM results in over-dissipation at small scales and fewer wavelet coefficients on the finest levels, which ultimately results in the earlier complete removal of the highest level of resolution from the adaptive computational grid, as clearly seen in Figs. 47 and 48. In contrast to the global model, the new models are capable of capturing the local structure of the flow, rather than providing only the mean energy dissipation.

We want to emphasize that, differently from classical LES, the SCALES solution matches the filtered DNS not only in terms of temporal evolution of the total resolved energy (or other global quantities), but also in terms of recovering the DNS energy and enstrophy spectra up to the dissipative wavenumber range. This close match is achieved using less than 0.5% of the total non-adaptive nodes required for a DNS calculation with the same wavelet solver. To highlight the significance of such an agreement, one can compare the present results as shown in Figs. 46 and 47 with those of 64^3 finite-difference non-adaptive LES supplied with either the global dynamic Smagorinsky model (as reported in (Vasilyev *et al.* 2006)) or the present energy-based ones. Despite the fact that LES solutions use about three times the number of modes, they fail to capture the small-scale features of the flow and the resolved kinetic energy spectrum is noticeably lower than the filtered DNS one for moderate and high wavenumbers. This leads to the under-estimation of the energy content of the flow-field so that the LES solutions appear over-dissipative for the first half of the simulation period as illustrated in Fig. 52, where the energy evolution and energy spectra for non-adaptive LES

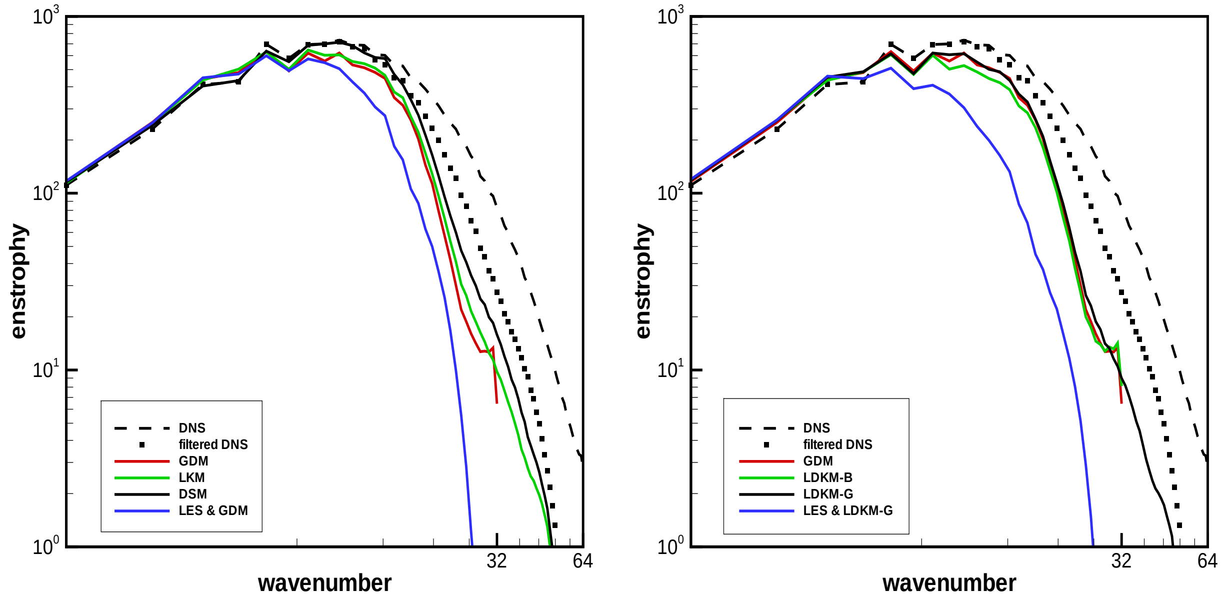


Figure 53: Enstrophy spectra at $t = 0.08$. Left side: DSM (———) and LKM (green line). Right side: LDKM-B (green) and LDKM-G (———). The reference GDM (red) and wavelet-filtered DNS (■) solutions are shown for comparison on both sides, along with the unfiltered DNS (– – – –). Furthermore, non-adaptive LES supplied with GDM and LDKM-G are reported (blue) on the left and the right side, respectively.

supplied with energy-based models are reported. These differences are more pronounced for the enstrophy spectra, which are illustrated in Fig. 53 (for $t = 0.08$). That is even more important since the enstrophy spectra, if properly normalized, coincide with the viscous dissipation spectra, so that the close agreement provided by wavelet-based adaptive LES ensures proper spectral distribution of resolved viscous dissipation.

Finally, it is instructive to discuss the “unexpected”, by classical LES standards, good performance of the dynamic structure model. In fact, in LES formulations, pure similarity models fail in providing the right SGS dissipation, leading to under-dissipative solutions, and therefore they require an eddy-viscosity model to be used in conjunction with them. In this work, the capability of the SCALES method to resolve small dissipative scales, at the small additional cost of slightly lower compression, makes it possible to avoid the use of an additional dissipative mechanism as in mixed formulations. Even though the present results are certainly affected by the low Reynolds-number nature of the flow, one can expect a similar good behavior to hold also for higher Reynolds-number simulations. That is one of the objectives of future work on the subject.

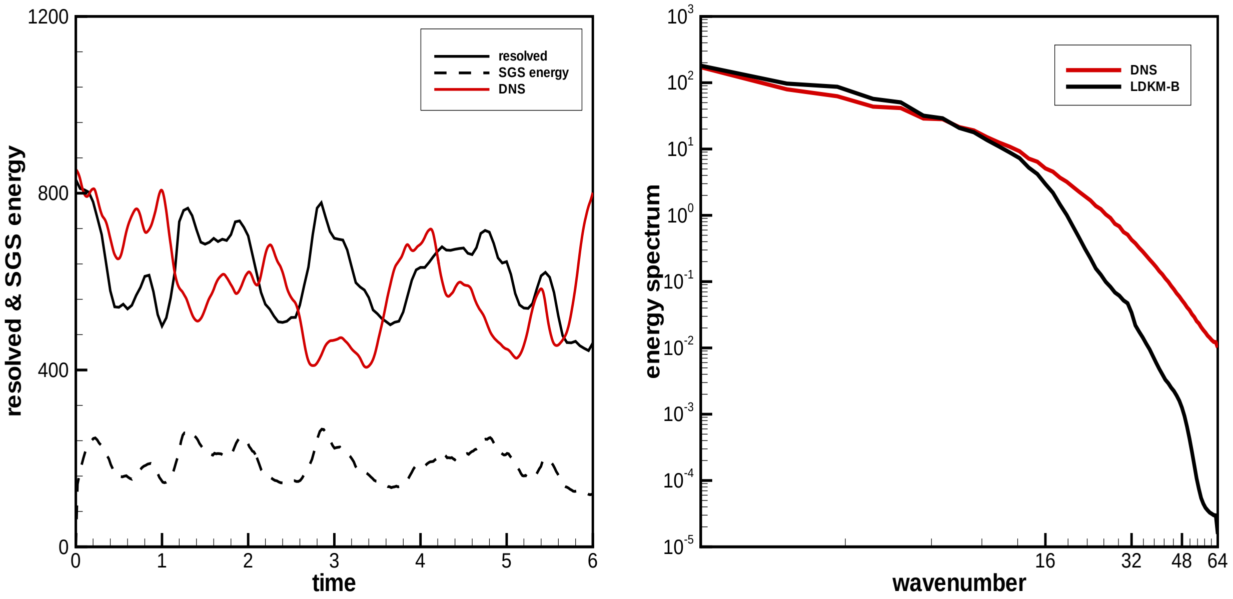


Figure 54: LDKM-B solution of forced turbulence. Left side: resolved (———) and SGS (- - - -) energy. Right side: averaged energy spectra. The reference DNS solution is reported on both sides (red line).

7.6 Forced turbulence

In order to test the energy-based modeling procedure for a statistically steady flow, let us consider the case in which a forcing term f_i is added at the right-hand-side of the momentum equation (12). Namely, following the linear forcing scheme proposed by Lundgren (Lundgren 2003), let the force be proportional to the velocity, $f_i = Q\overline{u_i}^{>\epsilon}$, Q being a constant parameter that can be determined from the energy balance corresponding to the steady state. This way, the solution is continuously supplied with the amount of energy necessary to keep the total resolved energy (statistically) constant in time. In fact, the parameter Q can be showed to be directly linked to the eddy turnover time of the turbulent velocity field (*e.g.*, (Rosales & Meneveau 2005)). For the present numerical experiments $Q = 6$ is used, which corresponds to have $\tau_{eddy} = 0.056$. The initial velocity field is obtained by wavelet-filtering of the fully de-aliased pseudo-spectral 128^3 DNS statistically steady solution with $Re_\lambda \cong 60$. The simulation is conducted for a temporal range of approximately one hundred eddy-turnover times.

The kinetic energy evolution for the LDKM-B solution of linearly forced homogeneous turbulence is illustrated, along with the reference unfiltered pseudo-spectral DNS, on the left side of Fig. 54. The corresponding time-averaged energy spectra are shown on the right side of the same figure. Once again, it is worth stressing how the wavelet-based solution is able to reproduce to some extent the energy of the small-scale motions. The SCALES solution shows a grid compression that is in average as high as 97%, while the SGS dissipation is

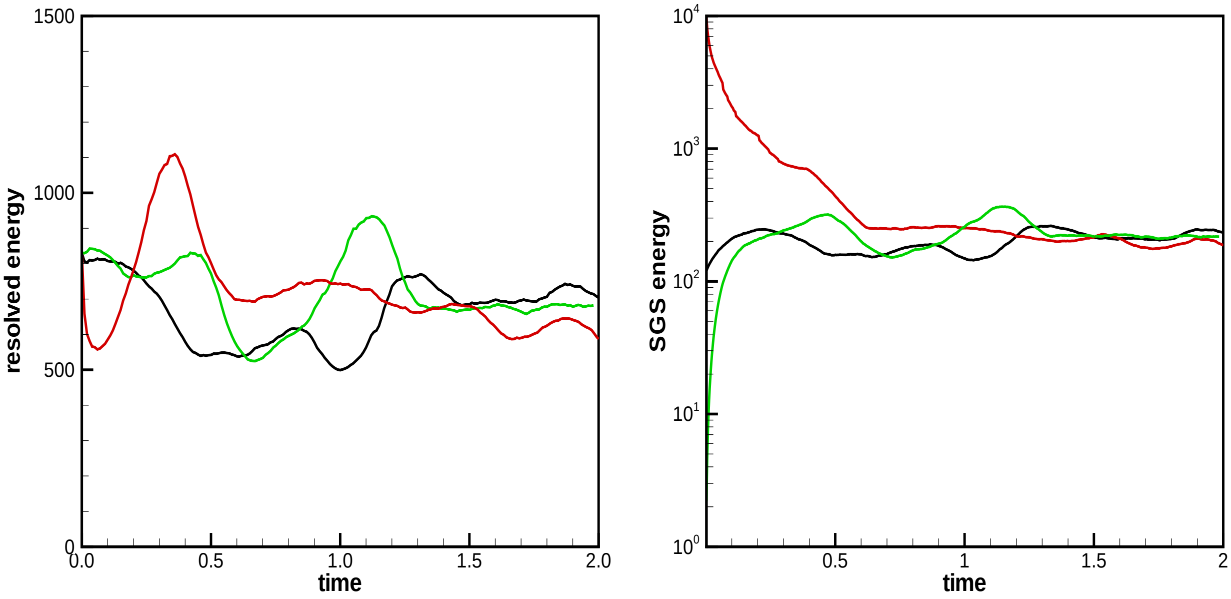


Figure 55: Energy evolutions for LDKM-B solution of forced turbulence with initial SGS energy altered by a factor of either 10^2 (red line) or 10^{-2} (green). The resolved and SGS energies are reported along with the unaltered solution (———) on the left and the right side, respectively.

of the same order of magnitude as the resolved viscous dissipation. Moreover, by properly redefining the Taylor microscale in terms of the total energy dissipation, resolved viscous dissipation plus SGS dissipation, the wavelet-based solution provides the same Reynolds-number as the reference DNS. These results demonstrate the effectiveness and efficiency of the energy-based SGS model in the forced case.

Finally, to definitely verify the stabilizing action of the built-in feedback mechanism associated with the dynamic energy-based modeling procedure, the following experiment is conducted: the initial SGS kinetic energy content of the flow is artificially altered by multiplying the variable $k_{\text{sgs}}(\boldsymbol{x}, 0)$ by a factor of either 10^2 or 10^{-2}. This way, the initial SGS energy is either much more or less than the equilibrium value provided by the wavelet-filtered DNS solution. Nevertheless, the LDKM procedure is able to provide a flow evolution that converges after some time toward the unaltered stable solution so that the equilibrium levels are restored. This is clearly illustrated by inspection of Fig. 55, where the evolutions of resolved and SGS energy are reported, on the left and the right side, respectively. This demonstrates that the energy-based method works in practice: solving a subgrid energy transport equation properly represents the energy transfer between resolved and SGS motions, both forward and backscatter.

8 Achievements and Future Extensions

This project has advanced the Adaptive Large Eddy Simulation methodology by developing multiple local subgrid scale models, implementing them in Stochastic Coherent Adaptive Large Eddy Simulation solver, and validating them in the context of homogeneous turbulence. The progress made in this project, both in the area of model and parallel algorithm development, has positioned us to take a next step and start applying the approach to more challenging problems of engineering interest, mainly high Reynolds number turbulent flows in complex geometries. To achieve this goal, the method needs to be extended to include Brinkman penalization and the models need to be tested, validated for bounded flows. Parallel Dynamically Adaptive Wavelet Collocation solver developed as a part of this project can be applied to other areas of physics and engineering, where localized structures play an important role. Obvious applications are in the area of chemical engineering, atmospheric sciences, material sciences, and bio-engineering.

The research program funded by the project provided opportunities for four graduate and one postdoctoral students to participate in multidisciplinary research and work together with a team of international researchers: Prof. Nicholas K.-R. Kevlahan (McMaster University, Canada), Prof. Giuliano De Stefano (Seconda Universitá di Napoli, Italy), and Dr. Daniel E. Goldstein (Northwest Research Associates). These students received research experience and training in fluid mechanics, wavelet theory, and numerical simulations. Two Ph.D. students involved in this project have graduated. The results of this research have been presented at professional meetings (APS, DLES6) and at universities and research centers around the world. The methodology and findings are documented in archival journals such as Physics of Fluids (De Stefano *et al.* 2008) and Journal of Turbulence (Goldstein *et al.* 2005, Vasilyev *et al.* 2008) as well as in conference proceedings (De Stefano *et al.* 2005, Vasilyev *et al.* 2006).

References

Angot, P., Bruneau, C.-H., & Fabrie, P., 1999, A penalization method to take into account obstacles in viscous flows, *Numerische Mathematik*, **81**, 497–520.

Bardina, J., Ferziger, J. H., & Reynolds, W. C., 1983, Improved turbulence models based on large eddy simulation of homogeneous incompressible turbulence, Tech. rep., *Report TF-19*, Thermosciences Div., Dept. of Mech. Eng., Stanford University.

Chasnov, J. R., 1991, Simulation of the kolmogorov inertial subrange using an improved subgrid model, *Phys. Fluids A*, **3**, 188–200.

Chumakov, S. G. & Rutland, C. J., 2005, Dynamic structure subgrid-scale models for large eddy simulation, *Int. J. Numer. Meth. Fluids*, **47**, 911–923.

Daubechies, I., 1992, *Ten Lectures on Wavelets*, no. 61 in CBMS-NSF Series in Applied Mathematics, SIAM, Philadelphia.

De Stefano, G., Goldstein, D. E., & Vasilyev, O. V., 2005, On the role of sub-grid scale coherent modes in large eddy simulation, *Journal of Fluid Mechanics*, **525**, 263–274.

De Stefano, G., Goldstein, D. E., Vasilyev, O. V., & Kevlahan, N. K.-R., 2005, Towards lagrangian dynamic sgs model for scales of isotropic turbulence, in *Proceedings of the Ercoftac Workshop D-LES 6, Poitiers, France*, pp. –, Edited by , Springer.

De Stefano, G., Vasilyev, O., & Goldstein, D., 2008, Localized dynamic kinetic energy-based models for stochastic coherent adaptive large eddy simulation, *Phys. Fluids*, **20**(4), 045102.1–045102.14.

Donoho, D., 1993, Unconditional bases are optimal bases for data compression and for statistical estimation, *Appl. Comput. Harmon. Anal.*, **1**, 100–115.

Donoho, D. L., 1992, Interpolating wavelet transforms, Tech. Rep. 408, Department of Statistics, Stanford University.

Donoho, D. L., 1994, De-noising by soft-thresholding, *IEEE Trans. Inf. Theory*, **41**(3), 613–627.

Durbin, P. A. & Reif, B. A. P., 2001, *Statistical Theory and Modeling for Turbulent Flows*, Wiley.

Farge, M., 1992, Wavelet transforms and their application to turbulence, *Annual Review of Fluid Mechanics*, **24**, 395–457.

Farge, M. & Schneider, K., 2001, Coherent vortex simulation (CVS), a semi-deterministic turbulence model using wavelets, *Flow, Turbulence and Combustion*, **66**, 393–426.

Farge, M., Schneider, K., & Kevlahan, N., 1999, Non-Gaussianity and coherent vortex simulation for two-dimensional turbulence using an adaptive orthogonal wavelet basis, *Phys. Fluids.*, **11**(8), 2187–2201.

Farge, M., Pellegrino, G., & Schneider, K., 2001, Coherent vortex extraction in 3d turbulent flows using orthogonal wavelets, *Physical Review Letters*, **87**(5).

Gatski, T. B., Hussaini, M. Y., & Lumley, J. L., 1996, *Simulation and Modeling of Turbulent Flows*, Oxford.

Germano, M., Piomelli, U., Moin, P., & Cabot, W., 1991, A dynamic subgrid-scale eddy viscosity model, *Phys. Fluids A*, **3**(7), 1760–1765.

Ghosal, S., Lund, T. S., Moin, P., & Akselvoll, K., 1995, A dynamic localization model for large-eddy simulation of turbulent flows, *J. Fluid Mech.*, **286**, 229–255.

Goldstein, D., Vasilyev, O., & Kevlahan, N.-R., 2005, CVS and SCALES simulation of 3D isotropic turbulences, *J. of Turbulence*, **6**(37), 1–20.

Goldstein, D. A., Vasilyev, O. V., & Kevlahan, N.-R., 2003, Feasibility study of an adaptive large eddy simulation method, *AIAA Paper 2003-3551*.

Goldstein, D. E. & Vasilyev, O. V., 2004, Stochastic coherent adaptive large eddy simulation method, *Phys. Fluids*, **16**(7), 2497–2513.

Goldstein, D. E., Vasilyev, O. V., Wray, A. A., & Rogallo, R. S., 2000, Evaluation of the use of second generation wavelets in the coherent vortex simulation approach, in *Proceedings of the 2000 Summer Program*, pp. 293–304, Center for Turbulence Research.

Guermond, J.-L. & Shen, J., 2003, Velocity-correction projection methods for incompressible flows, *To appear in SIAM J. Num. Anal.*.

Jameson, A., 2003, CFD for aerodynamic design and optimization: Its evolution over the last three decades, *AIAA Paper 2003-3438*.

Jimenez, J. & Moin, P., 1991, The minimal flow unit in near-wall turbulence, *J. Fluid Mech.*, **225**, 213–2406.

Jimenez, J., Wray, A., Saffman, P., & Rogallo, R., 1993, The structure of intense vorticity in isotropic turbulence, *J. Fluid Mech.*, **225**, 65–90.

Kaneda, Y., Ishihara, T., Yokokawa, M., Itakura, K., & Uno, A., 2003, Energy diisipation rate and energy spectrum in high resolution direct numerical simulations of turbulence in a periodic box, *Phys. Fluids.*, **15**, L21–L24.

Kevlahan, N. & Ghidaglia, J.-M., 2001, Computation of turbulent flow past an array of cylinders using a spectral method with Brinkman penalization, *Eur. J. Mech./B*, To appear.

Kevlahan, N. K.-R. & Vasilyev, O. V., 2005, An adaptive wavelet collocation method for fluid-structure interaction at high Reynolds numbers, *SIAM J. Sc. Comp.*, **26**(6), 1894–1915.

Kevlahan, N. K.-R., Vasilyev, O. V., Goldstein, D. E., & Jay, A. ., 2003, A three-dimensional adaptive wavelet method for fluid-structure interaction, in *Proceedings of Direct and Large-Eddy Simulation Workshop 5*, Technical University of Munich, Germany.

Khadra, K., Angot, P., Parneix, S., & Caltagirone, J. P., 2000, Fictitious domain approach for numerical modelling of navier-stokes equations, *Int. J. Num. Meth. Fluids*, **34**, 651–684.

Kim, W.-W. & Menon, S., 1999, An unsteady incompressible navier-stokes solver for large eddy simulation of turbulent flows, *Int. J. Numer. Meth. Fluids*, **31**, 983–1017.

Kravchenko, A. G., Moin, P., & Moser, R., 1996, Zonal embedded grids for numerical simulations of wall-bounded turbulent flows, *J. Comp. Phys.*, **127**, 412–423.

Lee-Rausch, E. M., Buning, P. G., Morrison, J. H., Park, M. A., Rivers, S. M., & Rumsey, C. L., 2003, CFD sensitivity analysis of a drag prediction workshop wing/body transport configuration, *AIAA Paper 2003-3400*.

Lesieur, M. & Métais, O., 1996, New trends in large-eddy simulations of turbulence, *Annu. Rev. Fluid Mech.*, **28**, 45–82.

Liandrat, J. & Tchamitchian, P., 1990, Resolution of the 1d regularized Burgers equation using a spatial wavelet approximation, Tech. rep., NASA Contractor Report 187480, ICASE Report 90-83, NASA Langley Research Center, Hampton VA 23665-5225.

Lilly, D. K., 1992, A proposed modification to the Germano subgrid-scale closure model, *Phys. Fluid*, **3**, 633–635.

Liu, S., Meneveau, C., & Katz, J., 1994, On the properties of similarity subgrid-scale models as deduced from measurements in a turbulent jet, *J. Fluid Mech.*, **275**, 83–119.

Lundgren, T., 2003, Linearly forced isotropic turbulence, *Annual Research Briefs*, pp. 461–473.

Mallat, S. G., 1999, *A Wavelet Tour of Signal Processing*, Academic Press, Paris.

Meneveau, C. & Katz, J., 2000, Scale-invariance and turbulence models for large-eddy simulation, *Annu. Rev. Fluid Mech.*, **32**, 1–32.

Meneveau, C., Lund, T. S., & Cabot, W. H., 1996, A Lagrangian dynamic subgrid-scale model of turbulence, *J. Fluid Mech.*, **319**, 353–385.

Moin, P., 2002, Advances in large eddy simulation methodology of complex flows, *Int. J. Heat Fluid Flow*, **23**, 710–720.

Moin, P. & Kim, J., 1997, Tackling turbulence with supercomputers, *Scientific American*, **January**.

Piomelli, U., 1999, Large-eddy simulation: achievements and challenges, *Prog. Aero. Sci.*, **35**, 335–362.

Pomraning, E. & Rutland, C. J., 2002, Dynamic one-equation nonviscosity large-eddy simulation model, *AIAA Journal*, **40**(4), 689–701.

Pope, S. B., 2000, *Turbulent Flows*, Cambridge University Press.

Pope, S. B., 2004, Ten questions concerning the large-eddy simulation of turbulent flows, *New J. Phys*, **6**(35).

Rosales, C. & Meneveau, C., 2005, Linear forcing in numerical simulations of isotropic turbulence: Physical space implementations and convergence properties, *Phys. Fluids*, **17**, 1–8.

Schneider, K., Farge, M., Pellegrino, G., & Rogers, M., 2003, CVS filtering of 3d turbulent mixing layers using orthogonal wavelets, Submitted to *J. Fluid Mech.*.

Schumann, U., 1975, Subgrid scale model for finite difference simualtions of turbulent flows in plane channels and annuli, *J. Comp. Phys.*, **18**, 376–404.

Smagorinsky, J. S., 1963, General circulation experiments with the primitive equations, *Mon. Weather Rev.*, **91**, 99–164.

Sweldens, W., 1996, The lifting scheme: A custom-design construction of biorthogonal wavelets, *Appl. Comput. Harmon. Anal.*, **3**(2), 186–200.

Sweldens, W., 1998, The lifting scheme: A construction of second generation wavelets, *SIAM J. Math. Anal.*, **29**(2), 511–546.

Vasilyev, O., De Stefano, G., Goldstein, D., & Kevlahan, N.-R., 2008, Lagrangian dynamic sgs model for stochastic coherent adaptive large eddy simulation, *J. of Turbulence*, **9**(11), 1–14.

Vasilyev, O. V., 2003, Solving multi-dimensional evolution problems with localized structures using second generation wavelets, *Int. J. Comp. Fluid Dyn.,* Special issue on High-resolution methods in Computational Fluid Dynamics, **17**(2), 151–168.

Vasilyev, O. V. & Bowman, C., 2000, Second generation wavelet collocation method for the solution of partial differential equations, *J. Comp. Phys.*, **165**, 660–693.

Vasilyev, O. V. & Kevlahan, N. K.-R., 2002, Hybrid wavelet collocation - Brinkman penalization method for complex geometry flows, *Int. J. Numerical Methods in Fluids*, **40**, 531–538.

Vasilyev, O. V. & Kevlahan, N. K.-R., 2005, An adaptive multilevel wavelet collocation method for elliptic problems, *J. Comp. Phys.*, **206**(2), 412–431.

Vasilyev, O. V. & Paolucci, S., 1997, A fast adaptive wavelet collocation algorithm for multi-dimensional PDEs, *J. Comput. Phys.*, **125**, 16–56.

Vasilyev, O. V., Yuen, D. A., & Paolucci, S., 1997, The solution of PDEs using wavelets, *Computers in Phys.*, **11**(5), 429–435.

Vasilyev, O. V., Podladchikov, Y. Y., & Yuen, D. A., 1998, Modeling of compaction driven flow in poro-viscoelastic medium using adaptive wavelet collocation method, *Geophys. Res. Lett.*, **25**(17), 3239–3242.

Vasilyev, O. V., Podladchikov, Y. Y., & Yuen, D. A., 2001, Modeling of viscoelastic plume-lithosphere interaction using adaptive multilevel wavelet collocation method, *Geophys. J. Int.*, **147**(3), 579–589.

Vasilyev, O. V., Goldstein, D. E., De Stefano, G., Bodony, D., You, D., & Shunn, L., 2006, Assessment of local dynamic subgrid-scale models for stochastic coherent adaptive large eddy simulation, in *Proceedings of the 2006 Summer Program*, pp. 139–150, Center for Turbulence Research.

Vincent, A. & Meneguzzi, M., 1991, The spacial structure and statistical properties of homogeneous turbulence, *J. Fluid Mech.*, **225**, 1–20.

Wang, M. & Moin, P., 2002, Dynamic wall modeling for large-eddy simulation of complex turbulent flows, *Phys. Fluids*, **14**, 2043–2051.

Yokokawa, M., Itakura, K., Uno, A., Ishihara, T., & Y.Kaneda, 2002, 16.4-Tflops direct numerical simulation of turbulence by a Fourier spectral method on the earth simulator, Tech. rep., http://www.sc-2002.org/paperpdfs/pap.pap273.pdf.